Context Analysis and Principles Study

Phase 1 - Client Report (Part 1) Architecture Practice Study

P. Chen and A. El-Sakka

DSTO-CR-0151

20010410 150

Context Analysis and Principles Study

Phase 1 - Client Report (Part 1)
Architecture Practice Study

P. Chen and A. El-Sakka

Joint Systems Branch
Electronics and Surveillance Research Laboratory

DSTO-CR-0151

ABSTRACT

Large organisations are becoming increasingly dependent on the reliability, adaptability and interoperability of information-intensive systems to conduct their business operations successfully and profitably. These qualities are now prerequisite for organisations' survival in the ever-changing national and international markets. For the last two decades, architecture, despite its diverse definitions, has been considered by Information Technology (IT) practitioners and researchers, as playing critical roles during the life cycle of systems and their infrastructure. The roles include, firstly, providing a sound foundation on which quality information-intensive systems are developed or evolved, secondly, capturing the necessary knowledge to aid in the understanding about these systems, and thirdly, guiding the development of enterprise infrastructure needed to support the creation of such knowledge. This report concludes that large organisations, including the Australian Defence Organisation (ADO), must commit themselves to establishing and managing architecture as an inseparable practice embedded within their overall IT management and practice. Such recommendation, once implemented, will ensure that the identified roles of architecture are performed, sustained and continually improved. To pave the way towards this objective, a conceptual model is introduced to help in the planning and guidance of the development and incorporation of architecture practice within ADO.

RELEASE LIMITATION
Approved for Public Release

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION **DSTO**

AQ FOI-06-1125

APPROVED FOR PUBLIC RELEASE

Context Analysis and Principles Study

Phase 1 – Client Report (Part 1)
Architecture Practice Study

Executive Summary

The main deliverable of Phase One of the Architecture Practice Study task is a technical report consisting of two parts. The first part is represented by this report, whereas the second part discusses specifically the requirements of IT development capability for the Australian Defence Organisation (ADO) and how the architecture practice introduced in the first part can support improvement of this capability.

This report captures research findings concerning the concept of architecture in IT, and its roles in large organisations. These roles of architecture include providing a sound foundation on which quality information-intensive systems are developed or evolved, capturing the necessary architectural descriptions to facilitate understanding about these systems, and guiding the development of enterprise infrastructure to support the creation of such descriptions. This report concludes that large organisations, including the Australian Defence organisation (ADO), must commit themselves to establishing and managing architecture as an inseparable practice embedded within their overall IT management and practice. Such recommendation, once implemented, will ensure that the identified roles of architecture are performed, sustained and continually improved.

The report highlights significant problems, more or less, common to current IT practice in large organisations. These include the high cost of acquiring and maintaining software (US DoD spends an estimated amount of US $30 Billion per year on software of which US $19.8 Billion is spent on software maintenance activities), the longer time to acquire and maintain software, inconsistent quality characteristics of acquired software (quality characteristics mean different things to different people in large organisations), and high turnover of IT professionals. The possible causes of these problems have been traced and analysed. The analysis discovered that IT practices in large organisations have disintegrated and immature capabilities preventing the effective generation, representation, preservation and retrieval of architectural descriptions in a consistent manner across the enterprise.

A broad consensus exists among IT/IS practitioners and researchers that architecture, despite its diverse definitions and meanings, is crucial to resolving these problems. Recognised and influential architecture-based frameworks of both industry and government organisations have been selected and studied. The study covers Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance - Architecture Framework (C4ISR AF), Microsoft Solutions Framework (MSF), Zachman Framework, Meta Group's Enterprise Architecture Strategies (EAS), The Open Group Architectural Framework (TOGAF), and Product Line Practice (PLP). The selected frameworks represent just a sample of the many attempts to eliminate

these problems. In all the selected frameworks, architecture has been central in their design.

In order to be able to provide a proper basis for reaching a better understanding of architectures and architecture frameworks or approaches, the report uses the whole architecture practice as the context to investigate different scenarios of architecture production and use, interrelationships and connections among various architectures and frameworks, and complexity of architecture practice in large organisations. Based on such a context, architecture practice is studied as a community practice in which professionals can communicate their architecture issues and products with less confusion and misunderstanding, and the value of each architecture product can be maximally realised. In the efforts to address architecture issues that are important for an emerging discipline but not addressed by most architecture frameworks, the report focuses on attributes of architecture and the knowledge value chain in architecture practice.

The study of the selected frameworks has shown strengths as well as weaknesses inherent in each of them. The main strength is found to be their ability to acknowledge the value of architecture for individual systems as well as for the enterprise as a whole, and to apply architecture thinking to manage architecture in their own environment as in the case of C4ISR AF. On the weakness side, two main limitations are common to each of these frameworks. The first limitation is in the missing link between the process of developing/evolving architectures for information-intensive systems and the enterprise architectural infrastructure. This missing link results in spreading incomplete and inconsistent architectural descriptions across the enterprise, thus causing unreliable systems and islands of information to flourish. The second limitation is in the absence of a process for codifying architecture descriptions prior to preserving them in an enterprise repository. This absence limits the use and reuse of architectural descriptions to specific areas in the organisation, thus preventing the benefits of these descriptions being shared by all areas of the enterprise.

The encountered limitations serve to demonstrate that none of the selected frameworks is capable, on its own, of providing a total solution to manage architecture in large organisations, including the ADO. A total solution here refers to an integrated solution that provides large organisations with the necessary capabilities to generate, represent, preserve and present architectural descriptions about information-intensive systems in a consistent manner across the enterprise with less time, cost and effort. Such capabilities have the potential to resolve IT practices' main problems and also overcome limitations described above.

This report introduces a new definition for architecture. It defines architecture of a system as knowledge regarding that system; the knowledge is described and represented by a set of interrelated views (models), which collectively reflect the concerns and requirements of the stakeholders of that system. Also, the report describes an "Architecture Practice Conceptual Model (APCM)", which addresses the limitations of existing architecture frameworks and paves the way towards establishing an architecture practice. The model is based on an integrated solution with the capabilities to generate, describe, represent and present architectural descriptions of information-intensive systems in a timely and consistent manner enterprise-wide.

The APCM consists of four integrated components. The "System Architecture Construction/Evolution Process (SACP)" implements the generation/evolution of architectural descriptions of information-intensive systems. The "Systems Architecture Acquisition Process (SAAP)" is a necessary process for codifying the graphical architectural representations generated by SACP prior to preserving it in an "Enterprise Architecture Repository (EAR)". The "Enterprise Supporting Elements (ESE)" provide the essential services as required by SACP in its capacity to guide the construction or evolution of new or existing architectures.

The report communicates to large organisations that one of the main objectives of establishing an architecture practice, as an emerging discipline, is the cycling and recycling of the architectural knowledge of systems within the enterprise, thus avoiding leakage of knowledge that can be essential to the survival of such organisations.

The report also introduces and explains two assessment models, which have been designed to allow large organisations including the ADO, using the proposed architecture practice, to assess the level of the architectural knowledge sustained by the practice. Architecture practice improvement guidance is also provided should organisations seek to further improve the level of their architecture practice. The two models are "Component Improvemevnt Model (CIM)" and "Knowledge Acquisition Improvement Model (KAIM)". The main purpose of CIM is to assess the improvement level of each of the four components of APCM, based on five specified improvement levels. KAIM is designed to assess the sophistication of the architectural knowledge produced by the practice as a whole; it has four improvement levels and is based on how well the four main components of APCM are integrated and managed within the practice.

Finally, the report identifies critical success and failure factors in the pursuit of guiding large organisations into what to adhere to and what to avoid as they embark on developing and implementing the proposed architecture practice. The main critical success factors include commitment by higher management to adopting and establishing such a practice and continued involvement and dedication by the practice's stakeholders to keep business and technology aligned. On the other hand, the critical failure factors include allowing vendors to take full responsibility for developing and implementing this practice, and lack of coordination in developing and supporting the practice's four main components.

In conclusion, this report lays the foundations for large organisations including the ADO to establish an architecture practice, as an inseparable part of IT management, based on developing and integrating the capabilities of the four components of the architecture practice conceptual model. The challenge will be to test this model in the ADO by engaging stakeholders concerned with the proposed architecture practice. It is through such a practice that the ADO can develop or evolve its information-based systems with reliability, adaptability and interoperability, thereby enabling the ADO to better develop new and evolve existing capabilities.

Authors

Pin Chen
Joint Systems Branch

Dr Pin Chen is a Senior Research Scientist in the Information Architectures Group of Joint Systems Branch and the task manager of the Architecture Practice Study (DSTO task JNT99/017). His current research interests include architecture practice study, software architecture, enterprise architecture, architecture methodologies, architecture supporting tools and environments, and architecture capability integration. Dr Chen joined DSTO in March 1996 after he completed his Ph. D. in Computer Science at the Australian National University.

Abdel El-Sakka
Joint Systems Branch

Mr Abdel El-Sakka received his Bachelor of Science Degree in Computer Technology (Honours) in 1979 from Teesside Polytechnic in the United Kingdom. He is currently studying his Master of Business Administration (MBA) at Deakin University through the Association of Professional Engineers, Scientists and Managers, Australia (APESMA). Mr El-Sakka spent over thirteen years working for the Commonwealth Department of Health. During that period, his work covered the development and maintenance of mainframe applications, investigation and experimentation with new and advanced computer technologies, and supporting and enhancing the practice for developing applications with emphasis on the application development methodology and application architecture. He joined Defence Science and Technology Organisation (DSTO) in May 1998, as a member in the Information Architecture Group (IAG) of the C3 Research Center in Fern Hill Park in Canberra. His research focuses on understanding the concept of "Architecture" and on researching how architecture can best be managed, as a practice, within the overall IT management and practice of large organisations, including the ADO. The main purpose of an "Architecture Practice" is to allow large organisations to acquire quality Information-Intensive Systems, and at the same time, capture the necessary knowledge about these systems. This captured knowledge can then be reused in full or in part in the development of future systems or in the evolution of existing ones.

Table of Contents

Tables

Figures

Abbreviations

ADO	Australian Defence Organisation
APCM	Architecture Practice Conceptual Model
API	Application Programming Interface(s)
AM	Application Model
APG	Architecture Planning Group
APSE	Architecture Practice Supporting Environment
ASD C3I	Assistant Secretary of Defence for Command, Control, Communications and Intelligence
AWG	Architecture Working Group
C4ISR AF	Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance Architecture Framework
CIA	Chief Information Architect
CIO	Chief Information Officer
COE	Common Operating Environment
DE	Development Environment
DIA	Denver International Airport
DPM	Design Process Model
DBMS	Database Management System
EAM	Enterprise Architecture Model
EAR	Enterprise Architecture Repository
EAS	Meta Group's Enterprise Architecture Strategies
EBA	Enterprise Business Architecture
EIA	Enterprise Information Architecture
ESE	Enterprise Supporting Elements
EWTA	Enterprise-Wide Technical Architecture
IAG	Information Architecture Group
IAP	Integrated Architecture Panel
IAWG	IEEE Architecture Working Group
IEEE	Institute of Electrical and Electronic Engineers
IS	Information System
JSB	Joint Systems Branch
LISI	Levels of Information System Interoperability
MSF	Microsoft Solutions Framework
PLP	Product Line Practice
PM	Process Model
RMM	Risk Management Model
SAAP	Systems Architecture Acquisition process
SACP	System Architecture Construction Process
SESC	Software Engineering Standards Committee
SIB	Standards Information Base
SNS	Security Network Services
SOS	System of Systems
TAFIM	Technical Architecture For Information Management
TM	Team Model
TOGAF	The Open Group Architectural Framework

Abbreviations (Cont.)

TS	Technical Specifications
UJTL	Universal Joint Technical List
ZF	Zachman Framework

1. Introduction

Architecture is a concept being adopted by both IT and business communities because of its ability to represent or describe knowledge about objects and/or systems. However, architecture in large organisations has been difficult to define, digest, communicate and implement since it is a term that is used broadly and inconsistently. This difficulty is caused by the fact that there are many issues associated with architecture, which need to work together. Some of these issues include: firstly, various types of architecture (data, functional, operational, business, application, technical, system, reference, middleware and so on); secondly, various levels of architecture (component, system, systems-of-systems and enterprise); and thirdly, various frameworks or methodologies for generation, management, coordination, use and evolution of architecture. These issues are relevant in various contexts and have to be consistent and even integrateable in order to achieve greater benefits from architecture. This complicated relationship among these issues contributes to the increasing complexity of architecture practice. No existing discipline or architectural methodology or framework addresses systematically and completely the relationship among these issues.

Successful architecture practice requires a discipline that can help large organisations develop their future capabilities and improve existing ones, through introducing systematic management of not only all architecture products but also all architecture-related activities. Despite the long history of using architecture, both explicitly and implicitly, architecture practice as a discipline is new to most large organisations. Many architectural methodologies or frameworks, in particular these so-called enterprise architectures, can be part of this discipline but are never able by themselves to serve as a complete solution for enterprise-wide architecture practice.

Architecture practice study as presented in this report is based on a context analysis of all architecture-related activities and their outcomes. The study uses an integrated architecture business cycle and an architecture value chain to encompass related architecture products, frameworks and activities or processes.

The problems with current IT practice and architecture practice, as identified in this report, stress that there is a critical need for a remedy or a solution to improve the effectiveness of the architecture practice in large organisations. This report focuses on the concept of architecture and its use as a critical element in successful development and evolution of individual and joint information systems.

This report discusses a selection of architecture-based solutions attempted by both industry and government organisations, including Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance Architecture Framework (C4ISR AF), Microsoft Solutions Framework (MSF), Zachman Framework, Meta Group's Enterprise Architecture Strategies (EAS), The Open Group Architectural Framework (TOGAF), and Product Line Practice (PLP). The roles of these architectural approaches in architecture practice are also examined.

Unsatisfactory problem areas in large organisations' IT practice, despite increasing use of architecture and architectural approaches, dictate the need for a comprehensive solution with better exploitation of the concept of architecture. This report elaborates the concept of architecture, (providing a more complete analysis in terms of its nature, roles and context where it is generated, used and evolved) by using a high-level conceptual model of architecture practice. This report describes in detail the main components that constitute the conceptual model. By introducing the principles of architecture practice, this study helps people make distinction between unplanned and uncoordinated practice and disciplined and well-managed practice. It is suggested that disciplined and well-managed architecture should aim to generate an integrated architecture capability through developing an Architecture Practice Supporting Environment (APSE).

This report also discusses the architecture practice capabilities of its APCM's main components and how to measure their improvements, and offers general guidance into advancing from one improvement level to another.

Unlike some documents that describe specifically an architecture framework or approach, this report aims to provide its readers with a comprehensive analysis of architecture practice and a thorough investigation into the principles of this practice as an inseparable discipline of the IT management and practice of large organisations. This work aims to provide IT professionals, who have varying interests in architecture, with a context such that they can see the relevance among various architecture products, activities and architecture frameworks and find better ways to think and work together. Through a unifying conceptual model:

- Architecture practice organisers (such as Chief Information Officers [CIOs], Chief Information Architects [CIAs], IT managers and planners) can see how the architecture capability generated from a disciplined architecture practice can support operation of their IT business and the improvement of IT development capability as a whole.

- Architecture and system developers (such as architects, system integrators and system analysts) can understand better how their work can be supported by architecture services generated from a disciplined and well-planned architecture practice and how their products can benefit other stakeholders.

- Researchers and other users of architectures can find out how their interests in architecture are related to various concepts used in architecture and (architecture) products generated. This will enable them to both obtain maximum benefit from the architecture resources and services, and help integrate their work as part of the practice. They can contribute by developing added architecture capabilities (such as architecture-based simulation, modelling and planning) to the APSE or making their products available to support the architecture practice.

Certainly, it is not the intention of the authors to repeat efforts made by architecture framework developers in addressing how to develop a specific architecture product step by step. The main motivation of the study is to provide support for large organisations that have found traditional architecture activities and individual

architectural approaches cannot meet the needs of increasing architecture use and are looking for better solutions to achieve an integrated architecture capability for their future development.

This report, the Part I of Client Report (Phase 1) of Architecture Practice Study, is a general study of architecture practice. Part II addresses specific issues of architecture practice for the Australian Defence Organisation (ADO).

2. Main Problems in Current IT Practice and Future Organisation Development

Building large information systems is a complex undertaking with a high degree of risk and uncertainty. There are many problems associated with current IT practice in large organisations. What we are interested in here is the main problems that are common to all IT practices. To gain better understanding of these common main problems requires identification of their symptoms (or effects) and thorough investigation and analysis of the underlying causes of these identified symptoms.

2.1 Common Symptoms

The common symptoms that an IT practice suffers from constitute real obstacles in allowing the practice to perform effectively its roles in the development, evolution and support of individual and joint information systems. Poor performance of IT practice in large organisations will have negative effects on the ability of the organisation to achieve its vision. Common symptoms include:

- High cost of acquiring and maintaining software.
- Extended periods for acquisition and maintenance of software.
- Inconsistency in the quality characteristics of acquired software.
- High turnover of IT professionals.

2.1.1 High Cost of Acquiring and Maintaining Software

To best explain this symptom, it is helpful to refer to the Software Architecture of the Guidelines for Successful Acquisition and Management of Software Intensive Systems - Appendix G [US DoD, 1996]. We focus here on two areas: software life cycle cost distribution and software maintenance activities.

Software Life Cycle Cost Distribution

The guidelines report that US DoD spends an estimated US $30 Billion per year on software. Figure 2-1 shows how this amount is distributed throughout the life cycle of software. Broken down in dollar terms, various life cycle activities' cost as follows:

- Requirement definition activities amount to US $0.9 Billion per year.
- Design activities amount to US $2.7 Billion per year.
- Implementation activities amount to US $2.1 Billion per year.
- Testing activities amounts to US $4.5 Billion per year.
- Maintenance activities amount to US $19.8 Billion per year.

From the above, one can note that maintenance activities account for two-thirds of the total cost of software per year.
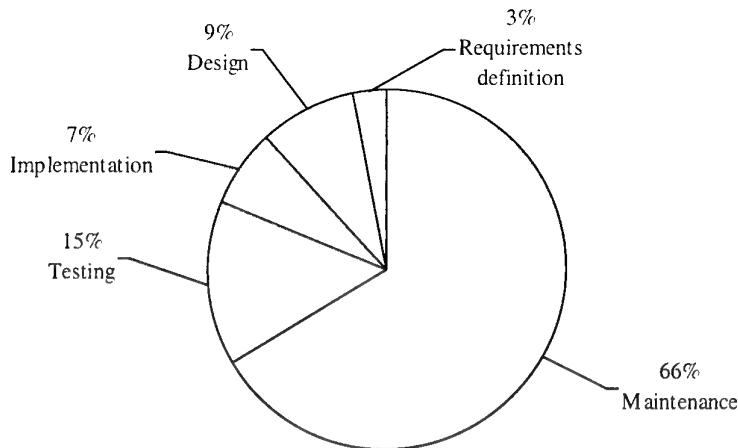
Figure 2-1 Software Life Cycle Cost Distribution

II. Software Maintenance Activities

The focus here is on software maintenance activities. Figure 2-2 shows how the yearly maintenance cost of US $19.8 Billion is distributed within maintenance activities. Below is a break down, in dollar terms, of the amount spent on each activity:
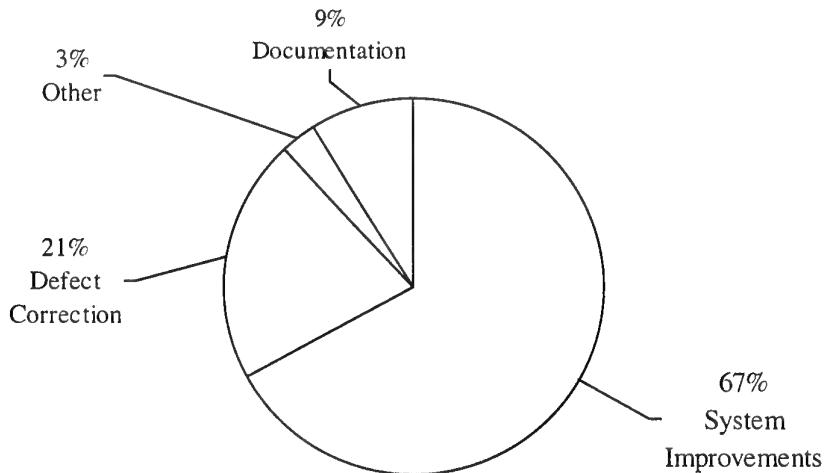
Figure 2-2 Software Maintenance Activities

- Cost of documentation amounts to US $1.8 Billion per year (9% of Maintenance Cost).
- Cost of system improvements (evolution) amounts to US $13.2 Billion per year (67% of Maintenance Cost).

- Cost of defect correction amounts to US \$4.2 Billion per year (21% of Maintenance Cost).
- Cost of other activities amount to US \$0.5 Billion per year (3% of Maintenance Cost).

What one can conclude from the cost distribution in the two figures above is that about 45% of the effort is spent on making changes to systems after they have been delivered. The US DoD needs US \$13.2 Billion per year to be able to change their existing systems to correct errors and provide them with improved functionality.

This brief analysis of cost distribution of both the software life cycle and software maintenance activities has generality; it is not only applicable to US but also applicable to large organisations world wide including the ADO.

2.1.2 Delays in Acquisition and Maintenance of Software

Government organisations as well as private organisations suffer from this symptom. In government organisations, the IT practice fails, on many occasions, to deliver new information systems in response to new government initiatives or legislation on time. Also, IT practice in public organisations fail to implement changes on time to existing information systems in response to changes in government requirements or legislation or to enhance functionality.

As for private organisations, current IT practice is equally responsible for not being able to deliver new or changed information systems in response to new business opportunities or to changes in existing business processes. One real life example of this symptom was the delay in delivery of the automated baggage handling system at the new Denver International Airport (DIA) in USA. This delay caused the rescheduling of the opening date of DIA from January 1, 1994 to February 1995. The delayed baggage system was attributed to significant mechanical and software problems, which the IT practice at the new airport failed to avoid. If DIA had opened by January 1, 1994, and operated in accordance with its final 1994 budget, the airport system would have incurred a \$37 million surplus instead of a \$230 million deficit through February 1995 [New Denver Airport, 1994]. This means that the real costs from the delayed opening of DIA totalled about \$267 million by February 1995. About \$86 million for modifications to the baggage handling systems would have been avoided. Therefore, the true delay costs were about \$361 million.

2.1.3 Inconsistency of Quality Characteristics in Acquired Software

Two possible factors may lead to this symptom:

- Some large organisations do not have a quality characteristics standard to guide them as to what quality characteristics their new or evolved systems should possess.

- The quality characteristic standards that exist in large organisations may have not been well communicated to or understood by all staff concerned; this

misunderstanding becomes evident when quality characteristics mean different things to different people in given organisations.

Table 2-1 shows quality characteristics and sub-characteristics as identified in the ISO/IEC 9126 standard. Systems that are developed with such characteristics qualify to be classified as quality systems (ISO/IEC 9126).

Table 2-1: Quality Characteristics Identified in ISO/IEC 9126 Standard

No.	Main Quality Characteristics	Sub-characteristics
1.	Functionality	Suitability, Accuracy, Interoperability, Compliance, Security
2.	Reliability	Maturity, Fault Tolerance, Recoverability
3.	Usability	Understandability, Learnability, Operability
4.	Efficiency	Time Behaviour, Resource Behaviour
5.	Maintainability	Analysability, Changeability, Stability, Testability
6.	Portability	Adaptability, Installability, Conformance, Replaceability

2.1.4   High Turnover of IT Professionals

Job satisfaction is one strategy that organisations can pursue to minimise the turnover of their employees. IT professionals are no exception, they need to feel satisfied in their jobs if the organisation is to retain them. Financial reward, achievement recognition and professional training will definitely have positive impacts in making IT professionals satisfied.
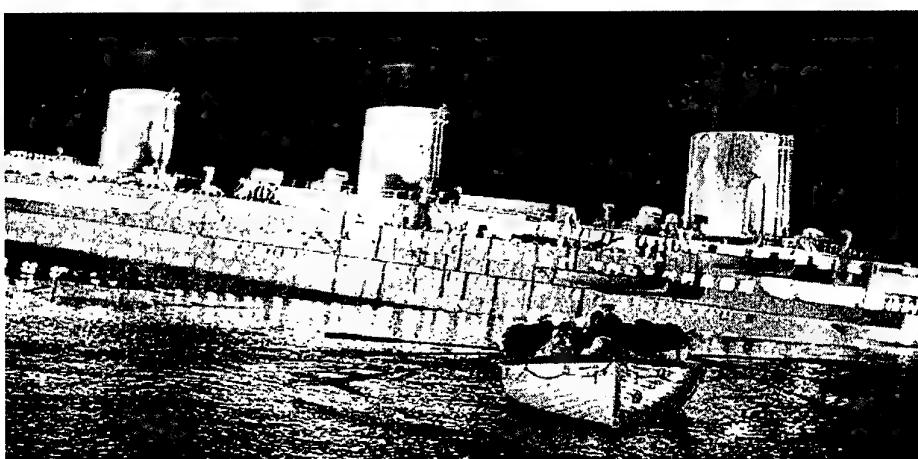
Figure 2-3 Immature IT Practice Likened to Sinking Titanic

7

Another factor, which can equally impact on IT professionals' satisfaction is the presence of a strategically sound IT practice in which they can conduct their activities in an effective manner and at the same time have the opportunity to participate in maturing the IT practice of their organisations.

In large organisations, the presence of immature IT practice that focuses on short-term goals would definitely lead IT professionals to lose faith and confidence in the ability of their organisations to charter their destiny. This loss of faith and confidence would greatly contribute to IT professionals leaving their organisations and causing a high turnover, and this is what Figure 2-3, portrays by likening an immature IT practice in a large organisation to the sinking Titanic.

According to Gartner Group [GG, 1997) from 1997 to 2003 the effective unemployment rate in the IT industry will be substantially negative globally; for every 10 full-time hires required, only 7.5 IT professionals will be available, albeit with regional variations (0.8 probability).

A direct consequence caused of high turnover of IT professional is great leakage of organisation knowledge, in particular the important knowledge generated in IT practice.

2.2 Causes

The possible causes of the symptoms of IT practice in individual organisations, discussed in Section 2.1, include but are not limited to the following:

- Inability to manage the increased complexity of architecture issues.
- Inability to manage the reuse of architectural descriptions of systems.
- Inability to manage the stakeholders' requirements/expectations.
- Inability to align IT and corporate business goals.
- Inability to provide enterprise-wide knowledge management solutions to prevent leakage of knowledge assets.
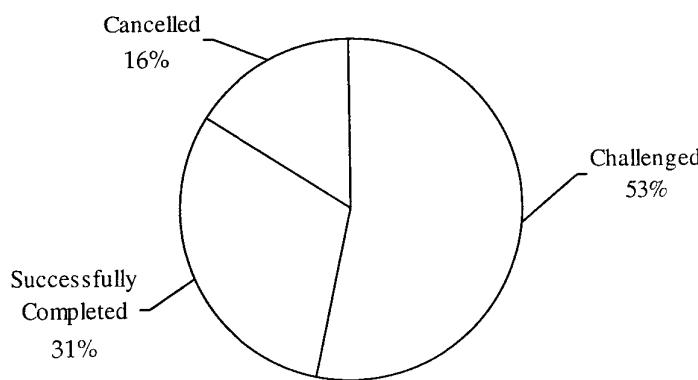
Figure 2-4 Results by Standish Group on the Success Rate of IT Projects

These inabilities directly contribute to the dissatisfaction of large organisations in their IT practice. As found by a study conducted by Standish Group in 1995, on the successes and failures of IT projects undertaken during the period from 1975 to 1995, only 16% of all IT projects were successfully completed in that they fully met the requirements of their stakeholders, and delivered projects on time and within the allocated budget. The study also found that 31% of all IT projects were impaired in that the projects were cancelled early in the development. Finally, they found that 53% of all IT projects were challenged in that they met 61% of the original requirements of their stakeholders, and delivered projects late and over the allocated budget (See Fig. 2-4). The cause here was that stakeholder's requirements and expectations were not fully met.

Remark

After three decades of practice in IT, large organisations face even greater difficulties and challenges in their future development due to inabilities in those aspects mentioned. It is time, thus, for large organisations to examine the level of their IT development capability or future development capability. This examination is necessary to determine whether or not these capabilities could meet the requirements of the organisation through delivering quality IT capability to support business, and how this IT development capability or future organisation development capability can be improved.

Through analysing the concept of architecture and investigating the principles of architecture practice, the remaining part of this report focuses on how a disciplined architecture practice can support improvement of IT development capability or future organisation development capability.

3. Concept and Use of Architecture in IT Practice

The symptoms and causes of the main problems in current IT practice, as described in Section 2, provide Information Technology/Information System (IT/IS) practitioners and researchers with a mission to find a long-term cure capable of eliminating these symptoms.

Some of the notable work done so far towards achieving this mission includes:

- Zachman Framework for Enterprise Architecture [Zachman, 1996].
- Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance Architecture Framework (C4ISR AF) [CAWG,1997b].
- Microsoft Solutions Framework (MSF) [Microsoft, 1999b].
- Meta Group's Enterprise Architecture Strategies [Meta Group, 1999].
- The Open Group Architectural Framework (TOGAF) [TOGAF, 1999).
- Product Line Practice.

The selected frameworks represent just a sample of many attempts to eliminate the symptoms outlined earlier. In all of these frameworks, the concept of architecture is central.

Reliance on the concept of architecture to develop these frameworks, is by itself a testimony of the growing importance of architecture as a foundation for the development and improvement of IT capability in large organisations. However, this architecture growth and its potential are making it more difficult to reach a commonly agreed answer to the question "What does architecture mean?".

This section gives a few of the many definitions available on architecture with the intention of showing the diversity of meanings and interpretations.

3.1 What Do Architecture Definitions Tell Us?

In the context of IT/IS, there is no shortage of definitions of architecture. The abundance of these definitions, which seem to lack commonality, contributes to the existing state of confusion, within and among organisations, on its meaning, and may also contribute to possible erosion of its importance.

The definitions listed below indicate the importance, attention, effort, and confusion generated by the promising concept called "architecture":

- IEEE

Architecture is the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time (IEEE Standard 610.12).

- John Zachman [Zachman, 1996]

Architecture is that set of design artifacts, or descriptive representations, that are relevant for describing an object such that it can be produced to requirements (quality) as well as maintained over the period of its useful life (change).

- Microsoft [Microsoft, 1999a]

Architecture is a general term referring to the structure of all or part of a computer system. Also covers the design of system software, such as the operating system, as well as referring to the combination of hardware and basic software that links machines on a computer network.

- Software Engineering Institute [SEI, 1999]

An architecture is a description of system structures, of which there are several (data flow, modular, process and so on). Architecture is the first artifact that can be analysed to determine how well its quality attributes are being achieved, and it also serves as the project blueprint. An architecture is also a description of the relationships among components and connectors. These are the things we mean when we use the word Architecture.

- Computer, Communications, Command, Control, Intelligence, Surveillance and Reconnaissance (C4ISR) [CAWG, 1997a]

An architecture is "the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time." (Same as IEEE definition)

- Object Ideas Corporation [OIC, 1997]

Architecture is a formal plan that guides aspects of business process automation that includes, but is not limited to; processes that are ideal for automation, designs that guide development, standards, etc ...

- Martin [Martin et al., 1994]

In its simplest form architecture is the planning and design work performed prior to the selection of an implementation technology. "Architecture involves deciding on patterns which systems will exhibit without restricting those decisions specific technologies or specific business processes."

Generally speaking, most definitions on architecture are clear and seem not to have any serious problems when they are used in the context where they are introduced. Situations often become, however, different and complicated when different architectures based on different definitions become relevant and have impact on each other within a complex context. Over-utilisation of the term "architecture" in IT and other related disciplines has led to emotional reactions to the architecture concept, mixed with hopes, confusion and disappointments. In such a situation, it is difficult for an organisation to fruitfully exploit architecture practice. *Achieving an acceptable and proper context-based explanation of "architecture" is the first critical issue faced by a large organisation in its architecture practice.*

11

3.2 IEEE's Recommended Practice for Architectural Description

The IT professional community will no doubt continue their attempts to define and explore the concept of architecture as this concept has not been consistently defined and applied within the life cycle of information or software-intensive systems. By studying the architecture definitions in Section 3.1, one can not escape how much they vary in definition, and diverge in meaning. These definitions become even more confusing when other terms are associated with Architecture such as Enterprise Architecture, Software Architecture, Systems Architecture, Operational Architecture and Technical Architecture. These variations can lead to confusion and may prevent common understanding from being achieved.

Despite significant industrial and research activity in the architecture area, there is no single, accepted framework for codifying architectural thinking into architectural description. The Institute of Electrical and Electronic Engineers (IEEE) has devoted much time and effort into examining the concept of architecture and its attributes.

As part of IEEE's effort, it formed the Architecture Planning Group (APG) in August 95. The work of APG encouraged the IEEE Software Engineering Standards Committee (SESC) to task APG with setting directions for incorporating architectural thinking into IEEE standards.

In April 1996, SESC created the Architecture Working Group (AWG) to implement the task goals recommended by APG. The *Recommended Practice for Architectural Description* [IAWG, 1998] was the first product of the Architecture Working Group. This recommended practice establishes a conceptual model for architectural description as shown in Figure 3-1, which summarises the key concepts introduced by this standard and their inter-relationships. The figure presents these concepts in the context of an architecture for a particular system, and an associated architectural description.

The model in Figure 3-1 introduces a set of terms that can be considered as attributes of the architecture entity. The terms are system, mission, environment, architecture description, rationale, stakeholder, concern, viewpoint, library viewpoint, view and model. These terms are briefly described below [IAWG, 1998]:

- A *System* is a collection of components organised to accomplish a specific function or set of functions. The term system here encompasses individual applications, systems in the traditional sense, sub-systems, systems of systems, product lines, product families, whole enterprises and other aggregations of interest.

- A *mission* is a use or operation for which a system is intended by one or more stakeholders to meet some set of objectives. A system exists to fulfil one or more missions in its environment.
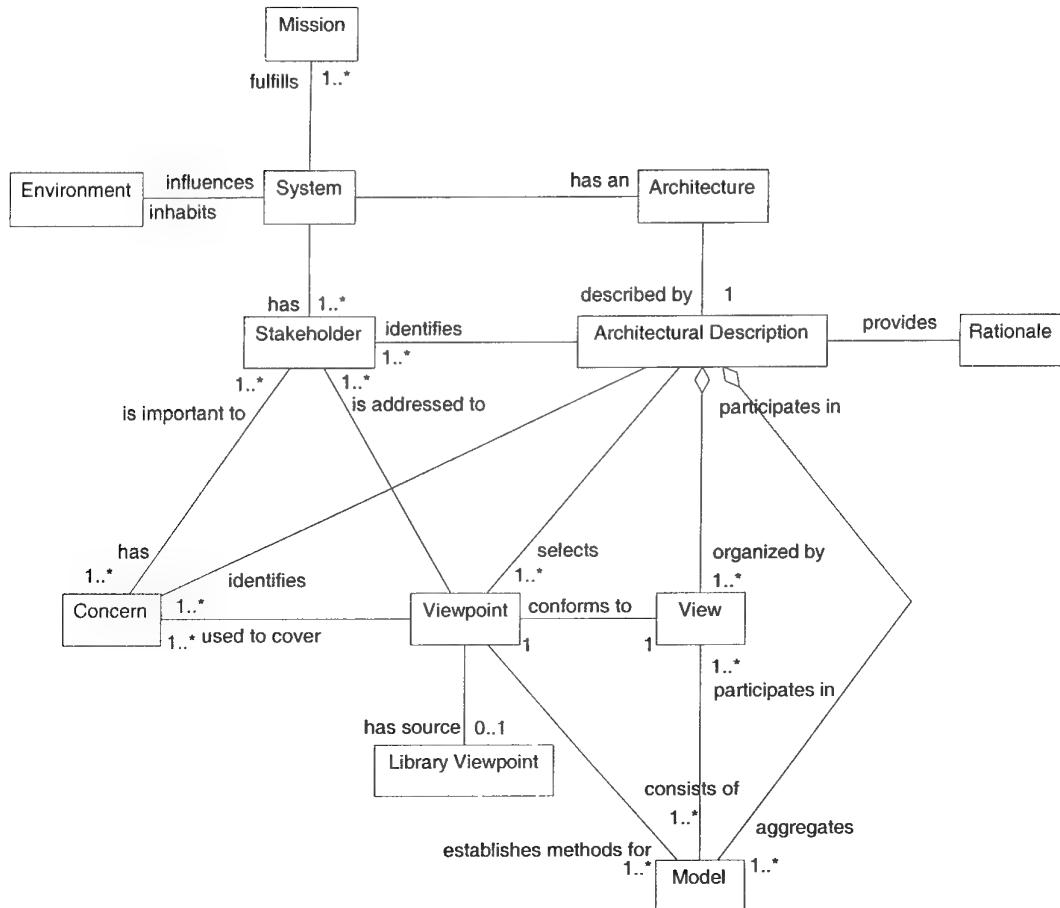
Figure 3-1 Conceptual Model of Architectural Description

- An *Environment* (or *context*) determines the setting and circumstances of developmental, operational, political and other influences upon that system. A system inhabits an environment, which in turn influences that system. An environment can include other systems that interact with the system of interest, either directly via interfaces or indirectly in other ways. The environment establishes the boundaries that define the scope of the system of interest relative to other systems.

- An *architecture* is the highest-level conception of a system in its environment. Every system has an architecture, whether or not that architecture is documented in an architectural description.

- An *architectural description* is a collection of products to document an architecture. An architecture can be recorded by an architectural description. An architectural description is organised into one or more (architectural) views. An *architectural description* will provide a rationale for decisions made about the subject architecture.

13

- A *view* addresses one or more concerns of the system stakeholders. The term "view" is used to refer to the expression of a system's architecture with respect to a particular viewpoint. A view conforms to a viewpoint.

- A *viewpoint* establishes the conventions by which the view is depicted and the particular architectural techniques or methods employed to create and document that view. In this way, a view conforms to a viewpoint.

- A system's *stakeholders* are those individuals, teams, and organisations (or classes thereof) with interests in, or concerns relative to, that system. A system has one or more stakeholders.

- *Concerns* are those interests, which pertain to the development, operation or other characteristics of the system, which are critical or otherwise important to one or more stakeholders. (Concerns include system characteristics such as performance, reliability, security, distribution, evolvability, etc.)

- An *architectural model* is part of a view. Each such architectural model is developed using the methods established by its associated architectural viewpoint.

- A *library viewpoint* is a previously defined viewpoint.

It is observed from the conceptual model in Figure 3-1 that architecture is always associated with a system, which relevant stakeholders use to facilitate the communication of knowledge. Stakeholders' concerns about a system can be properly described through introducing specific views into the architecture.

The conceptual model of architecture description provides an insight into the nature of architecture practice in the context where there is only one thing called "architecture". The reality of the practice, however, is that it is still difficult to fully exploit architecture in a proper context through this model. The complexity of architecture practice is caused indeed by the fact that architecture serves multiple purposes for different stakeholders through use of different views. However, relationships between systems or a system and systems of systems are not well defined and cannot be observed clearly, in particular during evolutionary development processes.

3.3 Use of Architecture in IT Practice of Large Organisations

Architecture is traditionally seen as the representation of an object or "system" with a view or a set of views for a specific purpose, but the term "architecture is used too broadly in current IT practice. There are many different types of architectures or architecture-related concepts which include, for example, organisational architecture, functional architecture, operational architecture, technical architecture, network or communication architecture, system architecture, software architecture, reference model, reference architecture, architecture frameworks and so on. These architecture-related concepts are architecture descriptions but differ according to the viewpoints used, and have been widely used in various IT activities associated with many *ad hoc* methods or approaches.

An *architecture product*, often called "an architecture", is an instance of a particular architecture description for a specific system. Any specific case of using architecture as either inputs or outputs should be in one of the development scenarios discussed in Section 5.2.

According to different objectives of R&D projects, their outcomes, or architecture products, can be quite different due to differences of systems facing them. However, they may be related to each other within given contexts of an organisation. Difficulties and problems in systems planning, design, implementation, maintenance, integration and migration are often caused by confusion and misunderstanding due to lack of model for clarifying and relating these concepts and activities.

Concurrent engineering, research and development activities in large organisations require a rational framework to relate various architecture products created or used in various activities. What would this framework be? How could we develop and use it in practice? What discipline should we use to develop such a framework? There have as yet been no adequate answers for these questions.

4. Overview of Architecture Frameworks/Approaches

An architectural framework is developed to guide, speed up and simplify architecture development, ensure more complete coverage of the designed solution, and make certain that the architecture selected allows for future growth in response to the needs of the business. This section provides an overview of some influential architecture-based solutions attempted by both industry and government organisations.

4.1 C4ISR AF

The Department of Defence, USA (US DoD), a foremost government organisation, acknowledges the importance of architecture as a critical element in the successful development and evolution of individual and joint software and software-intensive systems.

In October 1995, the Assistant Secretary of Defence directed that a DOD-wide effort be undertaken "...to define and develop better means and processes for ensuring that C4I capabilities meet the needs of the war fighters." To accomplish this goal, a C4ISR Integration Task Force (ITF) was established under the direction of the Assistant Secretary of Defence for Command, Control, Communications and Intelligence (ASD[C3I]). The Integrated Architectures Panel (IAP) of that ITF developed the C4ISR Architecture Framework to provide a basis from which the community could work collectively to evolve and mature architecture development concepts and promulgate them as DOD direction [CAWG, 1997a].

There are four main components that constitute the C4ISR Architecture Framework. The main components are Common Definitions, Common Building Blocks, Universal Guidance, and Common Products and Data.

4.1.1 Common Definitions

The "Common Definitions" component in C4ISR AF is designed to provide three views for an architecture and their logical interrelationship. The three views are Operational View, Systems View and Technical View.

- *Operational View* is a description of the tasks and activities, operational elements/nodes, and the information flows required to accomplish or support a military operation and fulfil mission obligations.

- *Systems View* is a description, including graphics, of systems and interconnections providing for, or supporting, war fighting functions and capabilities. A Systems View associates physical resources and their performance attributes to the operational view.

- *Technical View* is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements, whose purpose is to ensure that a conformant system satisfies a specific set of requirements.

4.1.2 Common Building Blocks

The "Common Building Blocks" component in C4ISR AF is designed to identify policies and documents that must be referenced in the process of developing an architecture. The common building blocks are the Universal Reference Resources (URR). These resources include:

- *C4ISR Core Architecture Data Model (CADM)* is the logical data model of information used to describe and build architectures.

- *Levels of Information Systems Interoperability (LISI)* is the Reference Model of interoperability levels and operational, systems, and technical architecture associations.

- *Universal Joint Task List (UJTL)* is the hierarchical listing of the tasks that can be performed by a joint military force.

- *Joint Operational Architecture (JOA)* (In development) is the High-level, evolving architecture depicting joint and multi-national operational relationships

- *Technical Reference Model (TRM)* is the common conceptual framework and vocabulary encompassing a representation of the information system domain.

- *DII Common Operating Environment (COE)* is the Framework for systems development encompassing systems architecture standards, software reuse, sharable data, interoperability and automated integration.

- *Shared Data Environment (SHADE)* is the strategy and mechanism for data sharing in the context of DII COE- compliant systems.

- *Joint Technical Architecture (JTA)* is an evolving set of elements consisting of services areas, interfaces, and standards that satisfy the DoD's technical architecture requirements.

Remark

It is worth noting here that while the above building blocks are part of the framework, but not owned by it as these building blocks are developed independently from the framework and can be used by different frameworks or approaches. The current set of these building blocks is not necessarily final since they are only those available when the framework was introduced. The evolution of these building blocks is a matter of great concern to the C4ISR framework as they may well influence the development and evolution of the three views.

4.1.3 Universal Guidance

The "Universal Guidance" component in C4ISR AF is designed to provide instructions to the architect, without prescribing specific tools or languages, for the purpose of developing an architecture. The universal guidance main instructions include:

- Determine the intended use of the architecture.
- Determine the architecture scope.
- Determine the characteristics to be captured.
- Determine views and products to be built.
- Build the requisite products.
- Use architecture for the intended purpose.

4.1.4 Common Products and Data

The "Common Products and Data" component in C4ISR AF is designed to provide a set of essential and supporting products to describe each of the three architectural views, namely operational, system and technical. To describe an architecture, the total number of products it requires will be as follows:

- For an operational view, 3 essential and 6 supporting products.
- For a systems view, 1 essential and 12 supporting products.
- For a technical view, 1 essential and 1 supporting products.

Remark

It should be noted that the typical use of the C4ISR AF is to develop C4ISR systems used for a *military operation*. This use can be explained as follows:

- The universal guidance is the starting point for the development of an architecture. The universal guidance helps the architect to determine the intended use and scope of the architecture and the views (Operational, Systems, and technical) and the products (Essential and Supporting) that need to be developed to describe the views.

- The architect and its team will use the architectural view templates (Common Definitions) to populate it using the Common Building Blocks as a references. As examples:

    – Commanders will reference the Universal Joint Task List (UJTL) in the "Common Building Block" to help them with task definitions as they construct operational view products.

    – Information Systems developers will reference the Level of Information System Interoperability (LISI) in the "Common Building Block" as they construct systems view products.

18

- The main output of the universal guidance is the architecture that embodies the three views, namely operational, systems and technical. Each of these views has its essential and supporting products, which should jointly serve the intended purposes.

Note that the C4ISR AF is an evolving approach and its current version is not perfect or even fulfilling the needs of the US DoD. A new version of the framework is currently being developed. It should be stressed, however, it aims to address the specific needs of the C4ISR domain.

4.2 Microsoft Solutions Framework (MSF)

Microsoft Solutions Framework is a holistic framework consisting of models, principles and practices aimed at aligning business and IT objectives in large organisations. MSF consists of six main modules as shown in Figure 4-1.

The modules are the Enterprise Architecture Model, the Team Model, the Process Model, the Risk Management Model, the Design Process Model and the Application Model.

Figure 4-1 shows the high level relationships and interdependencies among the six models of MSF. MSF is designed to expose critical risks, important planning assumptions and key interdependencies required for successfully planning, building, and managing a technology infrastructure or a business solution (Microsoft 1999[b]). The Enterprise Architecture Model plays a key role in supporting the alignment of IT development with business needs.
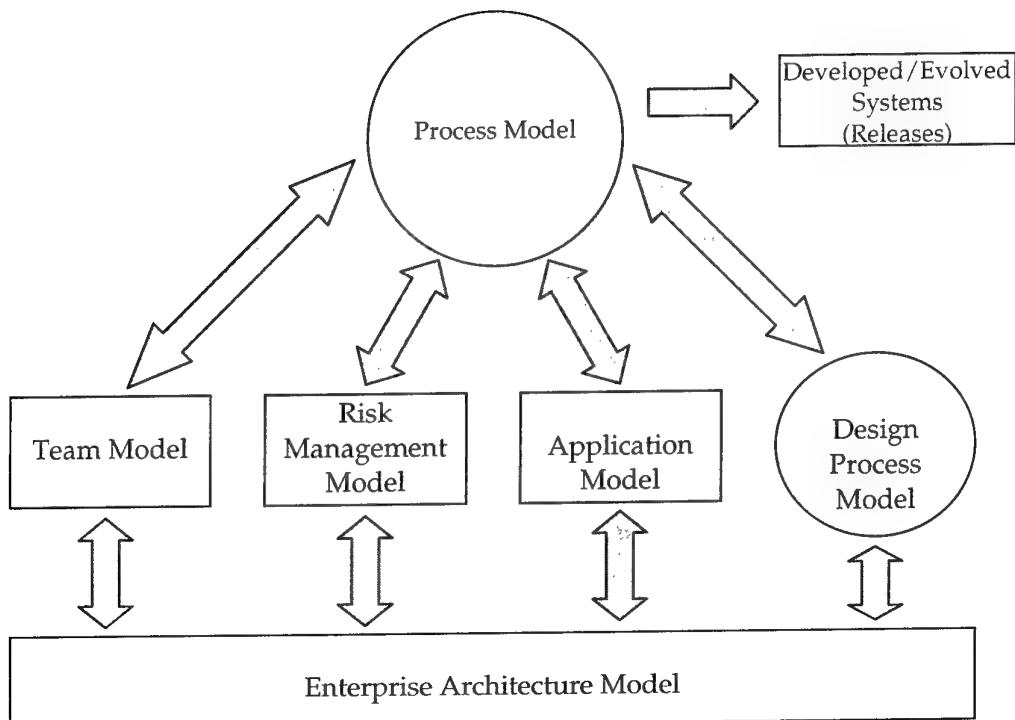
Figure 4-1 High Level Relationship of the Six Modules within MSF

When MSF is applied in large organisations, it aims to deliver the following benefits:

- Better IT infrastructure or business solutions developed in less time by fewer people.
- Building relationships between IT departments and users.
- Making smart tradeoffs between requirements, schedule and resources.

4.2.1   Enterprise Architecture Model (EAM)

EAM provides a consistent set of guidelines for rapidly building enterprise architecture through versioned releases. The model aligns IT with business requirements through four perspectives or architecture viewpoints: business, application, information and technology. Figure 4-2 shows the high level relationship of the four architectures within EAM.
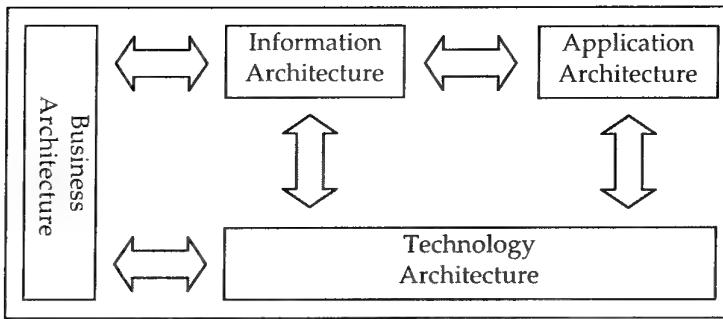
Figure 4-2 High Level Relationship of the Four Architectures within EAM

- Business Architecture

The business perspective describes how the business works in large organisations. It includes:

- The enterprise's high-level goals and objectives.
- The enterprise's products and services.
- The functions and the cross-functional activities the organisation performs embodied in business processes.
- Major organisational structures.
- The interaction of all these elements.

The business perspective includes broad business strategies along with plans for moving the organisation from its current state to its future state.

- Information Architecture

The information perspective describes what the organisation needs to know to run its business processes and operations. It includes:

- The standard data models.
- Data management policies.

The information perspective also describes how data is bound into the workflow, including structured data stores such as databases and unstructured data stores such as documents, spreadsheets and presentations that exist throughout the organisation. Often, the information most critical to an organisation resides not just in database servers, but on the thousands of desktop computers that comprise the enterprise's active working environment and in people's heads.

- Application Architecture

The application perspective defines the enterprise application portfolio. It includes:

- Descriptions of the automated services that support the business processes presented in the business architecture.

- Descriptions of the interaction and interdependencies (interfaces) of the organisation's application systems.

- Priorities for developing new applications and revising old applications based directly on the business architecture.

The application perspective represents the services, information, and functionality that cross organisational boundaries, linking users of different skills and functions to achieve common business objectives.

- Technology Architecture

The technology perspective lays out the hardware and software supporting the organization. It includes:

- Desktop and server hardware.
- Operating systems.
- Network connectivity components.
- Printers and Modems.
- Other necessary peripheral devices.

The technology perspective provides a logical, vendor-independent description of infrastructure and system components that is necessary to support the application and information perspectives. It defines the set of technology standards and services needed to execute the business mission.

These standards and services include, but are not limited to: Topologies, Development Environments (DE), Application Programming Interface(s) (API), Security, Network Services, Database Management System (DBMS) services, and Technical Specifications.

Although there are four perspectives, there is only one architecture. The value of the enterprise architecture is not in any one individual perspective but in the relationships, interactions, and dependencies among perspectives.

The enterprise architecture provides both information and a decision framework that enables IT management to arrive at a useable high-level plan. The plan includes both infrastructure and applications systems projects as well as providing standards, guidelines, and other support for the broad set of activities that must be accomplished to reach the desired state for the organisation. The enterprise architecture should be designed to evolve with the organisation, where modifications or upgrades directly mirror changing business needs and the application of new technologies.

Without losing the focus of our architecture discussion, we give only brief descriptions of the other five models of MSF.

4.2.2 Team Model

The Team Model provides a flexible structure for organising teams on projects. It emphasises clear roles, responsibilities and clear goals for team success, and increases team member accountability through its structure as a team of peers.

4.2.3 Process Model

The Process Model provides project structure and guidance through a project life cycle that is milestone based, iterative and flexible. It describes the phases, milestones, activities and deliverables of a project and their relationship to the Team Model. This model is designed to improve project control, minimise risk, improve quality, and shorten delivery time. It describes a life cycle that can be used for successful software development

4.2.4 Risk Management Model

The Risk Management Model provides a structured and proactive way for managing risk on projects. It sets forth a discipline and environment of proactive decisions and actions to assess continuously what can go wrong, determine what risks are important to deal with, and implement strategies to deal with those risks. Using this model and the principles and practices that underlie it will help teams focus on what is most important, make the right decisions and be better prepared for when the unknown future becomes known.

4.2.5 Design Process Model

The Design Process Model provides a three-phase user-centric continuum that allows for a parallel and iterative approach to design for the greatest efficiency and flexibility. The conceptual, logical, and physical design phases provide three different perspectives for the three different audiences: user, team and developers. Moving through conceptual design to physical design will show the translation of user-based scenarios to services-based components so that application features can be traced back to end-user requirements. Using this model helps ensure that applications are created not just for the sake of technology, but to meet business and use needs.

4.2.6 Application Model

The Application Model provides a logical three-tier services-based approach to designing and developing software applications. The use of user services, business services, and data services allow for parallel development, better use of technology, easier maintenance and support, and the greatest flexibility in distribution.

MSF emphasises the importance of using the Enterprise Architecture Model as a basis and combining it with other models in order to deliver quality IT capability serving better business needs.

4.3 Meta Group's Enterprise Architecture Strategies (EAS)

Meta Group structured EAS to attain two main goals. The first goal is to provide organisations with guidelines, templates and industry best practices for the purpose of designing and implementing an Enterprise-Wide Technical Architecture (EWTA). According to Meta group, EWTA is a logically consistent set of *principles* that guide the engineering of an organisation's ISs and technology infrastructure. The second goal of EAS is to provide organisations with operationally oriented guidance for ongoing assessment and optimisation of their EWTA. EAS consists of eight main components as shown in Figure 4-3.
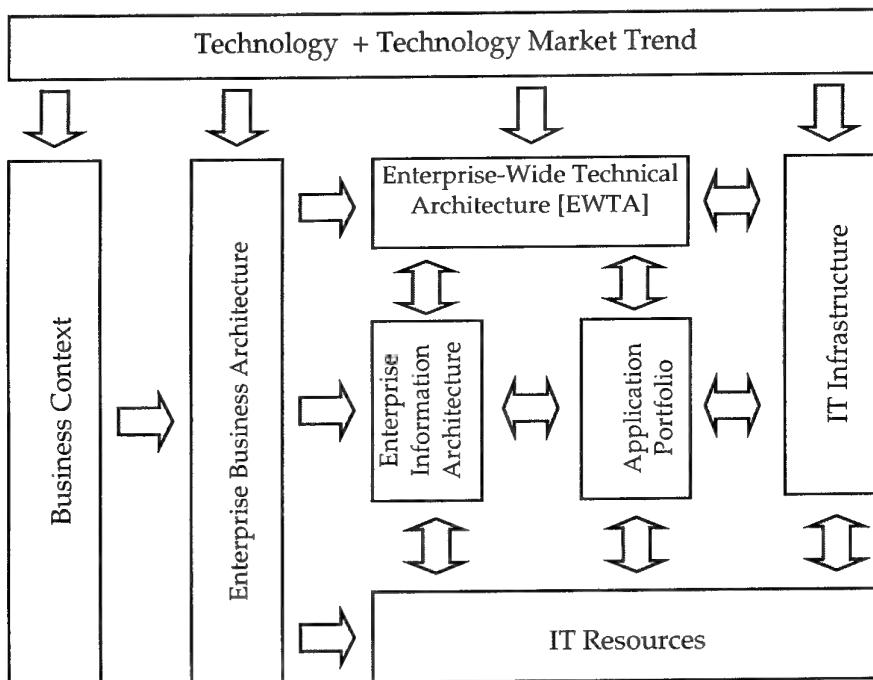
Figure 4-3   Main Components of Meta Group's EAS

4.3.1   Technology and Technology Market Trend

This component identifies the technology and trends in technology that will have an impact on the architecture strategy of an organisation.

4.3.2 Business Context

The business context relates to the business and environmental trend analysis of organisations and the response of these organisations to these trends in the form of enterprise business strategies.

4.3.3 Enterprise Business Architecture (EBA)

EBA defines the business events, owners and stakeholders of each of the business processes, as well as the interconnection of those processes, whether internal or external.

4.3.4 Enterprise Information Architecture (EIA)

EIA defines the flow of information between business events and business processes, whether internal or external.

4.3.5 Application Portfolio

An application portfolio is the collection of application solutions that enable and support the business processes described in the Enterprise Business Architecture (EBA). This component expresses what is or will be provided, not how it will be implemented.

4.3.6 Enterprise-Wide Technical Architecture (EWTA)

EWTA is a logically consistent set of principles that:

- Are derived from the business requirements.
- Guide the engineering of an organisation's ISs and technology infrastructure across the various domain architectures are understood and supported by senior corporate management and LOBs.
- Take into account the full context in which EWTA will be applied.
- Enable rapid change in a company's business processes and the applications that enable them.

EWTA consists mainly of a "Conceptual Architecture", which in turn decomposes into a set of "Domain Architectures". Each domain architecture is a set of design principles, international standards, product standards, and standard configurations for a collection of related technologies intended to guide the usage of those technologies.

4.3.7 IT Resources

This component consists of the skills, processes and organisational structures that will be needed for the development, evolution and support of EWTA. The organisation structures needed consists of IT Steering Committee, Architecture Review Board, Architecture Team, Domain Teams and Enterprise Program Management Office.

4.3.8 IT Infrastructure

IT infrastructure is the pre-existing technical foundation and resulting implementation of an EWTA. IT infrastructure includes not only the hardware, software, and communications used for computing and information sharing, but also the applications, data, process knowledge, repositories, methods, organisation, and staff/skill sets employed for computing and information.

4.3.9 EAS Process Phases

EAS process is designed to enable organisations attain two goals. The first goal is to create and implement EWTA, and the second goal is to guide ongoing assessment and optimisation of existing EWTAs.

EAS process consists of six main phases as briefly explained below:

- Phase 1

Develop a common vision between IT and the business on business drivers, business information requirements, requirements for architecture, and the impact of technology and technology market trends.

A business driver is a collective term used to describe the various influencing factors that cause (directly or indirectly) changes in an enterprise's business processes. The factors include external forces (changes in competition /market, globalisation, economy, customers, regulation, politics and technology) and responses to these forces (including business strategies, goals, objectives, requirements and strategies).

According to Meta Group [Meta Group, 1999], business strategy is defined as the planned responses to an enterprise's business drivers, intended to take advantage of opportunities presented by or to mitigate threats imposed by the business drivers.

- Phase 2

Define and document the conceptual architecture to provide logical consistency between the domain architectures in support of the requirements determined in Phase 1.

Conceptual architecture is a principles-based, enterprise-level layer of an EWTA to ensure clear decisions to sub-optimize individual domains to optimize total effectiveness of the overall EWTA to enable business strategies.

- Phase 3

Define and document the domain architectures (network, platform, middleware, data warehouse, e-commerce, application, etc..).

- Phase 4

Perform an analysis of the gaps between future and current state to determine the efforts necessary to migrate to the future state defined in Phases 2 and 3.

- Phase 5

Determine the priority, sequence and interdependencies of the efforts identified in Phase 4.

- Phase 6

Create detailed implementation plans for all migration efforts.

The procedure of going through these six phases of EAS shows an important feature of Meta Groups approach — a "top-down" philosophy of enterprise-wide architecture development for "to-be" systems, which starts with identifying enterprise business drivers. Such an approach may face difficulties when it is applied to a large organisation that is not in a position to re-develop most of its existing systems.

4.4 Zachman Framework

The Zachman Framework is a classification scheme for descriptive representations of any complex object (ie. enterprise). Zachman's philosophy is simply based on identifying the generic logic structure that organises, or classifies, the descriptive representations of complex objects. Objects here mean IS systems (The structure is shown in Figure 4-4).

Perspectives ↓	Abstractions →					
Scope (Contextual) *Planner*	**What** (Data)	**How** (Function)	**Where** (Network)	**Who** (People)	**When** (Time)	**Why** (Motivation)
Enterprise Model (Conceptual) *Owner*						
System Model (Logical) *Designer*						
Technology Model (Physical) *Builder*						
Detailed Representations (Out Of Context) *Sub-Contractor*						
Product Functioning *Enterprise*						

Figure 4-4 Zachman's Generic Classification Scheme

27

The rows represent five different perspectives: scope, owner, designer, builder and sub-contractor. The columns represent the abstractions: what, how, where, who, when, and why. Together, the perspectives and abstractions illustrate the complete "Enterprise Architecture". The key is the relationship between the perspectives and abstractions, whose intersection points provide the means to isolate and focus on a particular subset of the enterprise's architecture. The effects of a change on one intersection point can be gauged against the remaining points and the final effect on the organisation as a whole can be determined.

The Zachman framework provides an approach to identifying interests of different stakeholders by positioning their architecture products into a high-level representation for planning and management of enterprise-wide architecture development.

4.5 TOGAF - The Open Group Architectural Framework

TOGAF is an architectural framework that is a valuable tool for developing a broad range of different IT architectures. Most importantly, it enables developers to design, evaluate and build the right architecture for organisations. The key to TOGAF is a reliable, proven method — the TOGAF Architecture Development Method (ADM) — for developing an IT architecture that meets the needs of the organisation's business. TOGAF is focused on Architecture Management. It provides a rigorous discipline for developing and managing architectures.

The TOGAF Foundation Architecture — an architecture of generic services and functions that provides a foundation on which specific architectures and architectural building blocks can be built. This Foundation Architecture includes:

- The TOGAF Standards Information Base (SIB), a database of open industry standards that can be used to define the particular services and other components of an organisation-specific architecture.

- The TOGAF Architecture Development Method (ADM), which explains exactly how to get from the Foundation Architecture to an organisation-specific one. The ADM provides:

  - A reliable, proven way of developing the architecture.
  - Architecture views, which enable the architect to ensure that a complex set of requirements is adequately addressed.
  - A worked example and linkages to practical case studies.
  - Tools for architecture development.
  - Consulting services for those who need them.

4.6 Product Line Approach

The product line approach is a system of software production that uses software assets to modify, assemble, instantiate or generate a line of software products (ie building a software product line as a product family). A product line can be defined as a set of products sharing a common, managed set of features (ie. technology, design, parts and manufacturing process) that satisfy specific needs of a selected market or mission. Individual products may have different specific features and functionality required by different sets of customers.

Many familiar examples of product-lines exist in the manufacturing and retail areas. One of these examples is the automobile, in which companies use the same engines, transmissions, frames, factory infrastructure, etc., in different models of cars which are marketed for different purposes.

Put simply, product-line approaches allow higher quality products to be produced more quickly and with lower risk by leveraging proven components and processes. The product line approach conceptual model is shown in Figure 4-5.
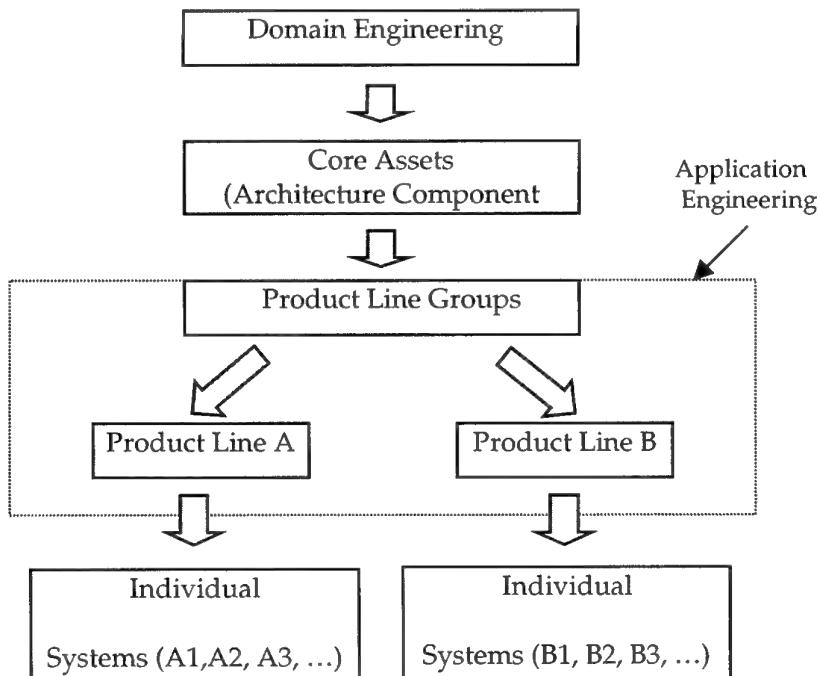

Figure 4-5 Product Line Approach Conceptual Model

- **Domain Engineering**

Domain engineering is the process used to create artefacts useful across the entire product line.

- **Core Assets**

Core asset is a software asset or other investments (such as training, estimates or work breakdown structure) that are used in multiple systems.

- **Product Line Application Groups**

Each of these groups is responsible for providing and sustaining product line vision. Also each group is responsible for the delivery of systems related to their product line.

- **Product Line (A, B, …)**

Whether product line A or B, as explained above a product line A is a group of products sharing a common, managed set of features that satisfy specific needs of a selected market or mission.

- **Application Engineering**

Application engineering is the process used to produce a single product (system) by adapting the domain–wide assets.

- **Individual Systems (A1, A2, …/ B1, B2, …)**

These are the individual systems that act as members of the product line family.

4.7. Problems and Issues in Use of Architecture Frameworks

4.7.1 Confusion in selecting and using architecture frameworks

Apart from the architecture frameworks mentioned above, there are also other architectural approaches that are either vendor-technology dependent or implementation-solution dependent. Examining these architecture frameworks or approaches reveals that although there are certain similarities among some of them, many approaches have been developed to address different issues facing organisations in different development scenarios and to support generation of different types of architecture products. We can observe that there seem fewer problems in selecting and using architectural approaches in a traditional project context where an application system is designed and implemented. In a non-traditional or complex development context or at the enterprise level, however, selecting and using architecture approaches faces difficulties (sometimes due to difficult and critical decisions on which framework should be chosen and how it should be applied). There is no general guidance on how to correctly select an architectural approach for the specific needs of an organisation. Existing architecture frameworks or approaches have not addressed this issue.

4.7.2 Questions on the value of architecture

The value of architecture varies depending on what architecture it is. Although the importance of enterprise architectures is stated clearly according to those architecture frameworks, there are still doubts on the value of these products due to unsatisfied returns generated from them in practice. On the other hand, though people acknowledge the value of architecture as a design of a system in the development process, whether and how such an architecture can maintain its value in the organisation after implementation is not clear. There are also other questions related to the value of architecture, such as, how can a proposed architecture development be evaluated, or how can the costs of architecture development be reduced?

4.7.3 Inability to Manage the Reuse of Architectural Descriptions of Systems

The collective knowledge of an organisation (ie. enterprise knowledge) constitutes the main source for the generation and evolution of systems and their architectural descriptions (knowledge). Architectural description of a system is critical and considered as refined knowledge about that system. The knowledge generated about a system will then need to be described, preserved and maintained before it is added to the existed enterprise knowledge and considered ready for future reuse.
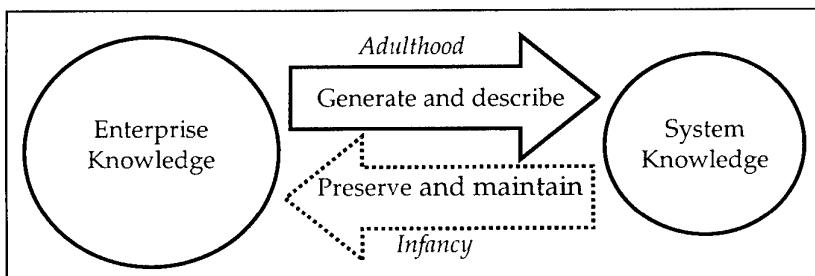
Figure 4-6 Knowledge Acquisition Cycle

Knowledge about a system or simply the architecture of a system goes through four stages or phases, namely, knowledge generation, knowledge description, knowledge preservation and knowledge maintenance. These phases constitute the system knowledge acquisition cycle or process. Most IT practice, in large organisations, is showing maturity in the first two stages of this cycle and is considered in its adulthood, as shown in Figure 4-6. At the same time, IT practice considered in its infancy in the other stages. This imbalance in maturation impacts on optimizing the reuse of the generated system knowledge or architectures of systems for the wider enterprise.

The inability to effectively manage the reuse of architecture descriptions of existing systems or to achieve a complete cycle of systems knowledge acquisition has made successful application of those architecture frameworks or approaches within large organisations more difficult.

5. Context Analysis of Architecture Practice

Architecture has been recognised and highly recommended as a useful concept and tool to improve IT practice. Organisations, in current IT practice, are not only faced with different methodologies in the way they develop their Information Systems but also are faced with a wide spectrum of architecture issues which differ from one organisation to another. Whether a large organisation can successfully make use of architecture in its future organisation development depends on how it can deal with the complexity of architecture issues among architecture products and activities, and generate best outcomes from them.

5.1 Areas of Architecture Issues

We classify architecture-related R&D activities into three sectors according to the features of their outcomes: architecture approaches, descriptive representations and development supporting environments, as shown in Figure 5-1.
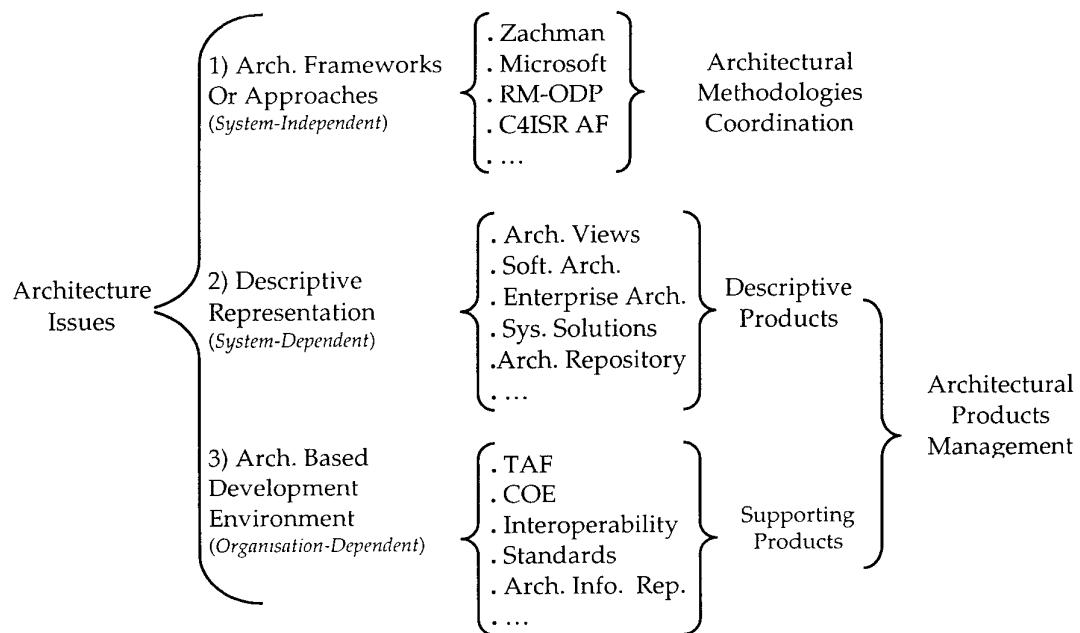
Figure 5-1 Architecture-Related R & D Activities

5.2 Use of Architecture in Different Development Scenarios

In order to understand when and where an architecture-related activity occurs and which architecture is generated for what purposes, there is a need to see clearly in which development scenarios the concept of architecture is used. This need intensifies

as the development context changes from supporting the development of a single system, to supporting the development of a single system in the context of system of systems, to supporting the development of a system of systems (SOS) in an evolutionary manner, as shown in Figure 5-2.
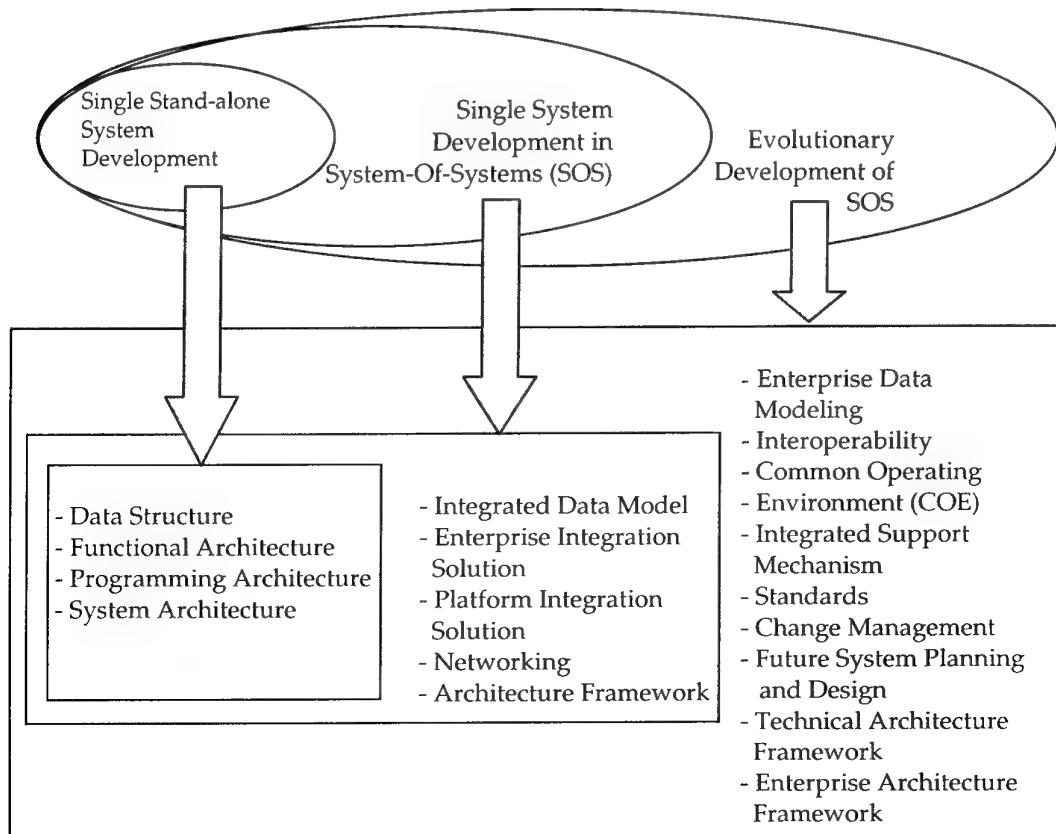
Figure 5-2 Increase in Complexity of Architecture Issues

5.2.1   Single and Stand-alone System Development

This type of system development is what is traditionally used to develop stand-alone systems irrespective of their computing platforms, where they are developed with no interaction with other systems. The architecture issues that need to support the development of such systems are usually data architecture, function architecture, software architecture, etc. These architectures are typically developed separately for each of these single systems. These single systems are usually stand-alone, they neither share data or functionality — they do not interface with other systems.

5.2.2   Single System Development in SOS Context

This type of system development is what is traditionally used to develop single systems irrespective of their computing platforms, where they can no longer be developed in isolation from other systems since they do interact with others. The architecture issues that need to support the development of such systems cover not

only the architecture issues of single systems but added architecture issues such as: integrated data model, enterprise integration solution, platform integration solution, networking, etc. The increase in the number and complexity of architecture issues becomes evident when such systems require interfacing with either the data or function of another system. In current IT practice, most development activities within a large organisation are carried out in such a setting.

5.2.3 Evolutionary Development of System of Systems (SOS)

A system of systems is a super-system, which consists of a number of components. Each component is a system in its own right. The components collectively constitute the super–system which can perform unique functions that the individual components cannot perform on their own.

This type of system development is the optimum goal organisations are set out to achieve, in particular when the entire organisation or enterprise is seen as a "big system" or system of systems. In that sense, the whole IT practice is the *evolutionary development of SOS*. The architecture issues that need to support the development of such systems cover not only the architecture issues of the previous two development scenarios, but additional architecture issues such as: enterprise data model, interoperability, Common Operating Environment (COE), standards, technical architecture, etc. The increased number and complexity of architecture issues is evident when such systems are required to communicate with other systems in the local and global domains with a high level of interoperability.

Architecture practice is a complicated practice within any large organisation if the architecture is broadly used to support various activities in future organisation development. For an organisation to be successful in architecture practice, the organisation needs to make correct decisions about the following:

- How can architecture products be developed?
- How can the products be used, managed, maintained and reused successfully?
- Which supporting elements should be developed?
- How can these elements evolve coherently along with changes in both business and technology?

Strategically, large organisations need to make decisions or find solutions to:

- Selection of architectural approaches.
- Coordination among architectural approaches if more than one is used.
- Coordination among processes that generate architectures.
- Management of architecture products generated in different processes through using different approaches.
- Strategic directions of architecture practice — main objectives.

Outcomes of architecture practice are architecture products and relevant supporting elements, which can jointly present certain capability to support management and future development of the organisation. The real value of the outcomes of architecture

lies in their use as validated *knowledge* in supporting IT practice. It is observed that most organisations started their architecture practice based on one of a few elements in the sectors shown in Figure 5.1, and then tried to extend to broader areas. The broader the practice, the more confusion arises. Much of the confusion can be attributed to a lack of context exploitation and investigation in architecture practice.

5.3 Complexity of Architecture Practice

Since architecture practice for most large organisations is not well planned and managed, the potential of architecture, through the concurrent use of many architectural approaches, is not being fully realised. This situation arises due to three common problems appearing in most architecture activities, that is, *incompleteness*, *inconsistency* and *confusion*. It is not surprising that some large organisations or certain business domains, like C4ISR, may face even more chaotic situations due to the increasing complexity of architecture issues.

The complexity of architecture practice increases as the practice covers more areas or involves additional activities. There are a number of factors that contribute to the increase in complexity.

5.3.1 Diversity of activities

Architecture issues addressed in different activities are classified as shown in Figure 5-1 into three main sectors in terms of their nature and outcomes.

5.3.2 Interrelationship amongst activities and outcomes

Detailed examination of architecture issues reveals that the increasing complexity is directly caused by interrelationships amongst architecture activities and their outcomes. Either the roles of individual architectures or contributions made by architecture activities have impacted on other products or activities in IT practice. There is no means to clarify such interrelationships in existing IT disciplines, such as Software Engineering and programming methodologies.

5.3.3 Continuity and concurrency of activities

Continuity or lifecycle of architecture activities varies according to their nature. Traditional architecture practice is characterized by an architect's work, ending before implementation commences. It is basically a project-wide practice. Some architecture products, such as technical architecture frameworks, however, require on-going efforts to maintain their validity. Concurrency of activities is evident as projects are usually carried out in parallel for SOS development. In addition, a given system theoretically has only one "as-is" architecture, but the business domain where the system is currently serving can have a number of "to-be" architectures, or simply the "to-be" architecture of the overall system. Thus, the time dimension further complicates architecture issues.

5.3.4   Number of people involved in the practice

Architecture practice is an on-going process of a community with people coming and going, perhaps working for different agencies and vendors for different projects. In such a community of practice, facilitating communications, knowledge sharing and integration, knowledge management and reuse, and knowledge preservation and evolution is methodologically essential for large organizations.

5.3.5   Mixture of static and dynamic products

Architecture activities of individual projects require limited management since they often produce their own products that are either static (such as design) or dynamic (living documents such as technical architecture frameworks). However, architecture practice needs a significant amount of organization-wide management and coordination of activities and products. Evolution of architecture products is another challenge for many individual architecture activities.

5.4   Use of Architecture Frameworks or Approaches

It should be acknowledged that each architecture framework or approach is developed to address certain architecture issues in a certain development setting discussed in previous sub-sections. Successful use of an architecture framework or approach requires a clear understanding of where and how it should be used. To reach such an understanding and avoid improper use, practitioners need to learn the framework or approach and also to examine the context/environment where it is going to be applied.

The investigation into the experience of using architecture frameworks shows that the degree of success largely depends on:

- Whether people can identify the right context to use the framework at the right time to generate the right products.
- How well it can be used in combination or jointly with efforts that address different issues in the architecture practice.

Note that any framework or approach tells usually only what it can deliver but does not tell clearly where it may not be suitable. As a mandated approach, for example, the C4ISR Architecture Framework is a military-operation-oriented approach for the C4ISR domain, in particular for a military mission context. Whether it is a good methodology for developing enterprise architectures or an infrastructure architecture, such as a network communication architecture, is questionable. Various problems have been observed in improper use of certain architecture frameworks or approaches.

There are some questions regarding architecture frameworks that are not addressed by the frameworks themselves. For instance, whether an organisation should only use a selected framework or approach or should use more than one? Why is a selected one

better than others? Whether and how can they be used together if multiple frameworks are adopted?

After analysing the nature (outcomes, scenarios of use) of architecture activities as shown in Figures 5-1 and 5-2 and the complexity of architecture practice, we not only realise the need but also approach a point that we must study the principles of the whole architecture practice. Such principles differ from principles of individual architecture frameworks or methodologies.

5.5 Principles of Architecture Practice

Architecture has traditionally been considered as "the art and science of building; structure; style of building; structure or building collectively; overall design of software and especially hardware of a computer or local network; organisation; framework" (Chambers Dictionary).

In other disciplines, such as civil engineering, architecture development and use is standardised in terms of notations and views. We note, however, that architecture development and use in IT practice is far from mature, in particular it is not yet used broadly by the community.

Architecture practice in the context of evolutionary development of SOS is a kind of "missing" science that is unfamiliar to people because of its integrative knowledge or transdisciplinarity across information systems, systems engineering, knowledge engineering and organisation study. It is regarded as a kind of intermediate discipline that bridges related disciplines and provides a common basis for their integration. Apart from dealing with architecture products, architecture practice also addresses issues related to methodologies or practice processes and practice supporting environments. Without a well-scoped architecture practice, organisations experience great difficulties in:

- Understanding the real world with its rapid changes in concepts and technology.
- Facilitating communications among people.
- Planning and implementing future development.
- Assessing various IT related systems and projects.

5.5.1 Architecture attributes and context

The reality of broadly using the concept of architecture necessitates a more comprehensive understanding of the concept. Based on the context analysis in previous sections, the nature of an architecture product is determined by its main attributes in the following aspects:

- *View type* (data, functional, business, reference, application, networking, enterprise and so on).
- *System/object associated* (component, system, systems-of-systems, enterprise).
- *Role* (descriptive or supporting/guiding — use of architecture).

- *Time dimension* (architecture associated with an existing system — "as-is" or a future system — "to-be").
- *Methodology-related* (representation features of architecture).
- *Tools-related* (form or format in which architecture is stored).

When the architecture is instantiated, it thus has multi-dimensional attributes as shown in Figure 5-3. An architecture product can be clearly understood and can be integrated with others only if all its attributes are well defined or specified throughout its lifecycle. It is these attributes that determine the context of a specific architecture (product), in which it is not only developed but also used. It is the responsibility of architecture developers to provide a clear context for their architecture products with specifications on those attributes. If the context is not clearly specified, the consequence is that the developers may have difficulty in communicating their work to the rest of community, restricting the value of their products.
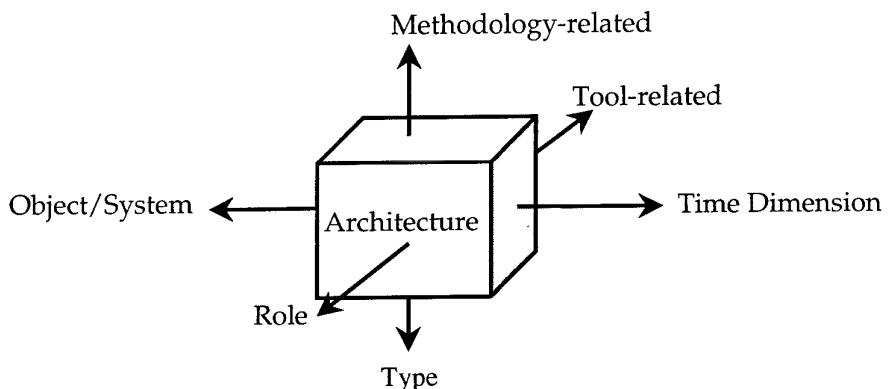
Figure 5-3 Typical Attributes of an Architecture Product

An architecture product can be either simple or comprehensive depending on its attributes. A simple architecture product can be either a descriptive product or a supporting/guiding product. Examples of simple architecture products are given in Figure 5-4, which can be expressed in any selected format. A comprehensive architecture product can be a combination of a set of simple architecture products.

It is important to recognize that architecture is not an automated process: it is the result of design, by people (architects), and needs management.

5.5.2   Architecture practice as a discipline

Architecture practice is initiated when various architecture products are generated in an organisation. This practice addresses well certain traditional architecture issues at the project level of application development. The problems and symptoms in current IT practice discussed in Section 2, however, have demand improvements in architecture practice. It should be considered and developed as an emerging discipline that aims to:

- Bring together related disciplines and addressing systematically principles of development, management and use of architecture.

- Address the issues that are not usually covered by individual frameworks or approaches.

- Achieve an integrated architecture capability for improvement of future development capability of the organisation.

Without such a discipline, an organisation can develop various individual architectures but it is hard to continuously and cost-effectively develop and maintain a successful and integrated architecture capability.

Types of Products

Either, *Descriptive Products* including:

↠ Design of system
↠ Structure/functional/behavior diagrams
↠ Solutions of system organisation/composition
↠ Activity models
↠ Business/process/data flow models
↠ Operational models
↠ Structural description of data / organisation

Or, *Supporting/Guiding Products* including:

✱ Standards
✱ References
✱ Rules
✱ Principles
✱ Guidelines
✱ Plans
✱ Technology options/trends
✱ Library
✱ Frameworks

Figure 5-4 Types of Architecture Products

Thus, the features of such architecture practice study can be summarised as:

- The study is about the context of architecture-related R&D. It covers architecture products, architecture methodologies and supporting environments.

- The objects of the study are relationships amongst various architectures of SOS and their development processes rather than any particular system based on a single-vendor solution.

- The evolutionary characteristics of the organisation and systems constantly add uncertainty to the context or environment of the practice.

- Continuity of the study with the organisation's evolutionary development process is necessary and important.

Principal goals of architecture practice study are to:

- Develop a proper classification of architecture products and associated methodologies.

- Investigate interrelationships and connections between different types of architectures.

- Seek effective solutions for architecture product management.

- Examine architectural approaches and their products.
- Explore inter-relationships among architecture-related R&D activities in IT in a unified development framework.
- Provide support to study the overall situation of organisation IT practice.
- Develop methods for evaluation of architecture products and frameworks.
- Guide and monitor architecture practice.
- Achieve an integrated architecture-based capability for future organisation development.

It must be noted that the above goals of architecture practice are unachievable through use of any single architecture concept or architectural approach. This is the main difference between architecture practice and any specific architectural framework or methodology. Architecture practice can, therefore, be seen as an inclusive discipline that claims to provide a rational context for supporting better development, management and use of architecture.

6. A Conceptual Model of Architecture Practice

The context and complexity analysis of architecture practice performed in the last section dictates the need for a comprehensive solution which consider the roles of architecture and ways in which these roles can be exploited. This section introduces a recommended architecture practice conceptual model that is built on these roles, to aid in the analysis of the context of an architecture product and an architecture framework. This section also discusses the capability of the recommended practice through its main components and proposed mechanism for assessing the practice maturity.

6.1 Exploitation of Architecture

The *Architecture* of a system or an object is defined as knowledge about that system or object. This knowledge is structurally represented and described by a set of interdependent views, which collectively reflect the concerns and requirements of the stakeholder community (Business and Technical) of that system (El-Sakka *et al*. 1999). A single view of a system can be described as an architecture (product) as far as it is given clearly in context, such as data architecture, network architecture and so on.

To better understand architecture in the context of architecture practice, it is important to examine it through the main roles it plays. Architecture can be viewed as having three distinct features/roles:

- Being a blueprint – basis for acquiring a new system.
- Being a current picture – basis for understanding an existing system.
- Being a roadmap – basis for guiding the development of the necessary infrastructure to support the first two roles.

A single and simple architecture can only partially play one of these roles at a given time. A set of architectures or a comprehensive one can jointly serve much broader roles. The great potential of architecture in supporting future organisation development can be achieved when all architectures can be developed, managed, used and evolved over time through well-organised architecture practice to jointly play the above three roles in supporting future organisation development.

6.1.1 The Blueprint Role of Architecture

Architecture's role as a blueprint is broadly and well accepted. The role of an architecture, as a blueprint in an ideal situation, is to converge the set of views, representing the requirements of the stakeholders, and turn it into a comprehensive architecture (knowledge about or designs of the planned system), which will form the basis for acquiring the future system.

Let us compare the set of views, representing the system stakeholders, with the wavelengths of light (Red, Orange, Yellow, Green, Blue, Indigo and Violet), and compare the architecture, as a blueprint, with a glass prism as shown in Figure 6-1.

If the glass prism receives the correct light wavelengths (seven colours), then the prism will converge them into a bright light, otherwise the quality of the convergent light will be degraded. Similarly, if the set of views is not fully representative of the system stakeholders, then the architecture will be incomplete, will not form a sound foundation for acquiring the future system and will not meet all requirements.

After implementation of a system, its architecture developed initially as a blueprint becomes explicit knowledge assets and then its role becomes that of a current picture. When a system undergoes change, the original blueprint may not remain a valid representation of the changed system.
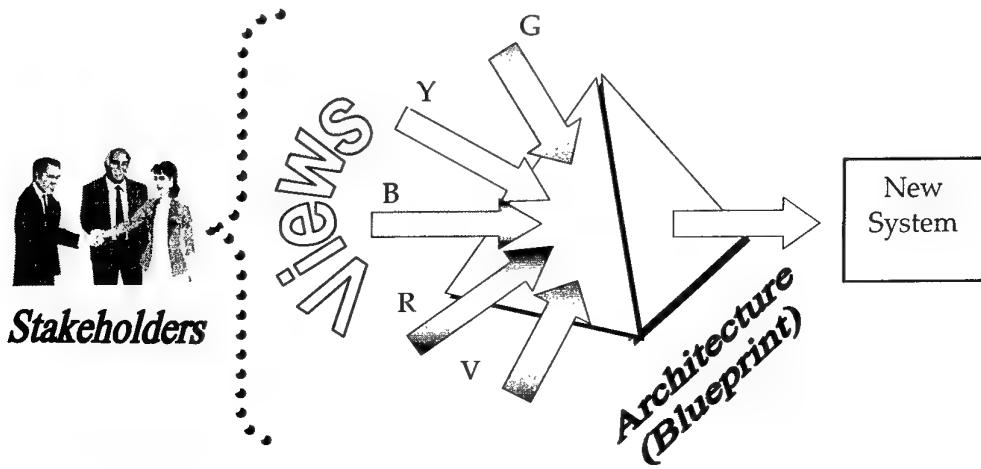
Figure 6-1 The Blueprint Role of Architecture

6.1.2   The Current-Picture Role of Architecture

The role of the architecture, as a current picture, is to present important knowledge of an existing system in a set of views, which collectively provide the necessary knowledge for each of the system stakeholders to understand the system from their own perspective.

Let us compare the set of views, representing the system stakeholders, with the wavelengths of light (Red, Orange, Yellow, Green, Blue, Indigo & Violet), and let us compare the architecture, as a current picture, with a glass prism as shown in Figure 6-2.

If the glass prism receives a bright white light, then the prism will diverge and produce the correct light wavelengths (the full 7 colours), otherwise the divergent light will be incomplete. Similarly, if the architecture of an existing system is representative of only some of the system stakeholders, then the architecture will not form a sound foundation for understanding the existing system.

When a system undergoes change, the current-picture will have to always remain valid and be a current representation of the changed system.
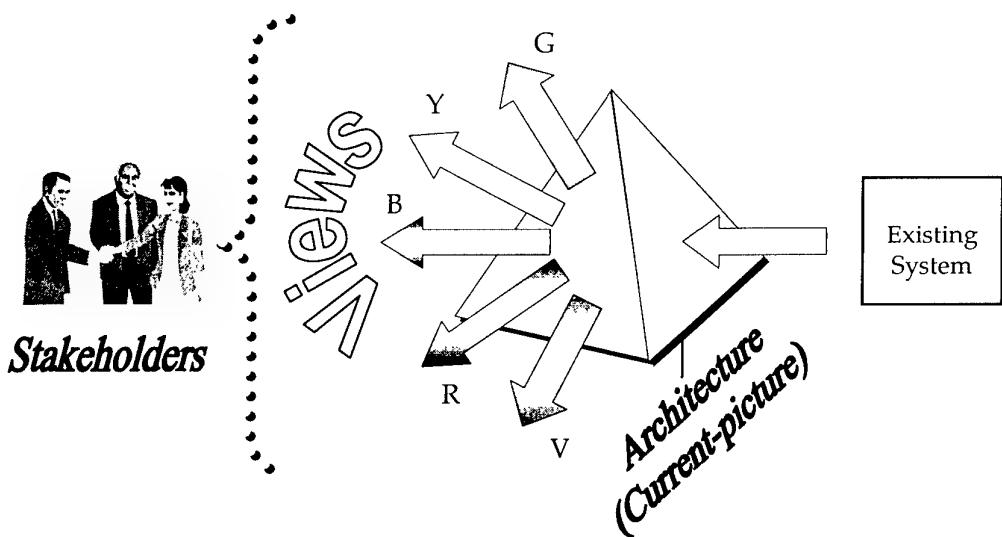
Figure 6-2 The Current-picture Role of Architecture

6.1.3 The Roadmap Role of Architecture

Unlike the blueprint and the current picture which are both descriptive products, the Roadmap Role of architecture is demonstrated by certain architecture products that provide support or guidance to the development of descriptive products. Such supporting roles of architecture are observed when in an enterprise context a technical architecture is developed to guide development of future system architecture such as with US DoD's TAFIM and EWTA of Meta Group.

Indeed, products that play roadmap or supporting roles are not necessarily architectures by themselves. They can be something related to architecture development. In the C4ISR AF, there are a number of such products called universal references or common building blocks.

6.2 Proposed Architecture Practice Conceptual Model

In order to examine all three roles played by architecture in the context of architecture practice, an architecture practice conceptual model is presented in Figure 6.3, which, as a high-level description of this emerging discipline, illustrates only main components. The model shows, first, how the system architecture is generated and evolves; how supporting architecture products are used, and then how an architecture (knowledge) value chain is formed. In another word, this proposed practice not only covers lifecycles of individual system architectures but also more importantly describes interrelationships among different architectures.

An important feature of the proposed model is that it is aimed at achieving a complete architecture knowledge value chain or integrated architecture business cycle. Based on such a knowledge value chain, capabilities developed associated with individual

43

architecture products can be integrated such that more of the power of architecture can be delivered in an integrated manner.
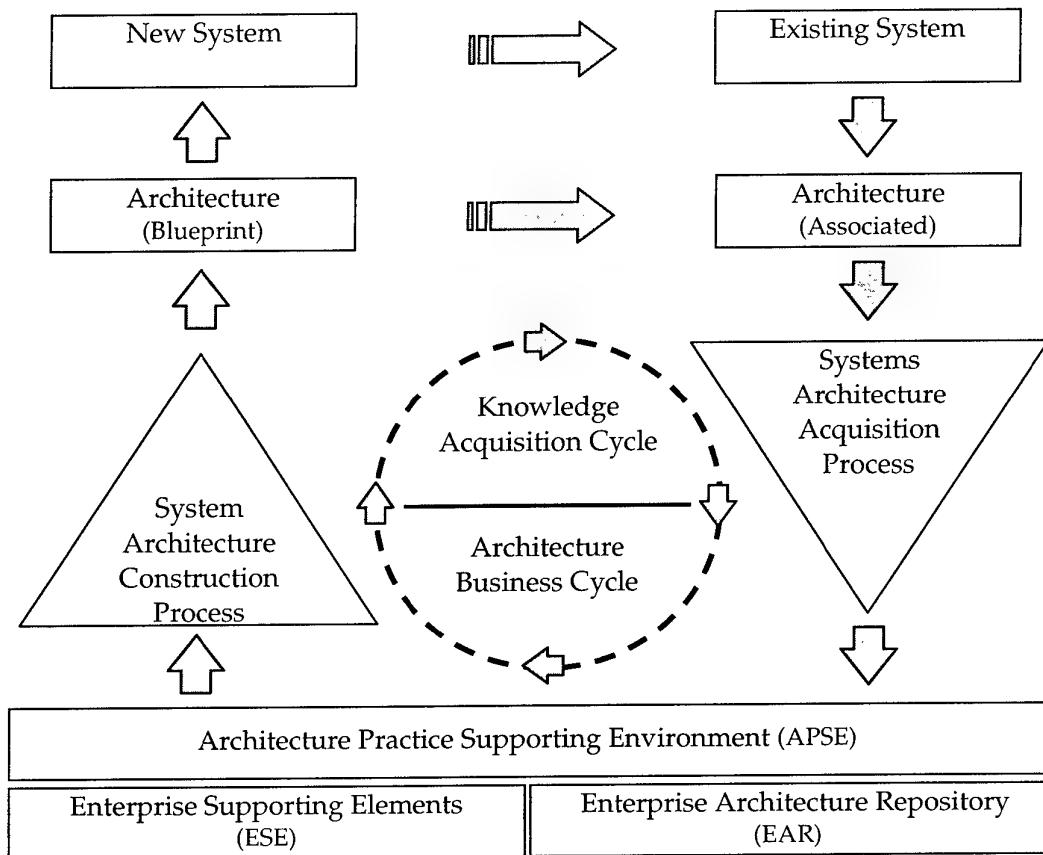
Figure 6-3 Proposed Architecture Practice Conceptual Model

It is important to point out that this proposed architecture practice model differs significantly from most architectural approaches or architecture frameworks. First, it covers much broader areas than those approaches or frameworks. Second, it focuses mainly on roles of architectures and their interrelationships rather than how a specific architecture should be developed. Through provision of such a context, finally, it can not only accommodate architectural approaches or architecture frameworks but also help them work together and to be integrated with other relevant architecture-related activities.

In addition, this proposed practice model could be used to support planning of high-level architecture practice for large organisations and to assist evaluation and selection of architecture approaches or architecture frameworks.

6.3 Capabilities Delivered by the Architecture Practice Model

The proposed architecture practice conceptual model of in Figure 6-3 is founded on three main pillars: System Architecture Construction process (SACP), System

Architecture Acquisition Process (SAAP) and Architecture Practice Supporting Environment (APSE). In turn, the APSE consists of two parts: the Enterprise Supporting Elements (ESE) and the Enterprise Architecture Repository (EAR).

The recommended architecture practice model has the potential to explore the capabilities associated with various architecture products. The main capabilities that can be observed in this model are:

- Services are provided by ESE to support system architecture generation through combination with adequate architectural approaches (such C4ISR AF).

- Architecture, as a blueprint, is generated by SACP to guide system development.

- Architecture, as a current picture, is acquired by SAAP to preserve the value of knowledge for reuse.

- Architecture resources including both ESE and descriptive products in EAR provide a (architecture) knowledge environment for developing advanced architecture-based or related capabilities, such as architecture analysis and simulation or modeling of SOS.

6.3.1 Role of the Enterprise Supporting Elements (ESE)

The number of elements that should be included in this component is largely dependent on which views describe the architecture of a system and how they should be constructed in SACP, in other words, using which architecture framework or approach. The elements in the ESE component are considered as reference resources.

As an example, when developing a new system, SACP is used to construct its blueprint. The system blueprint is the knowledge (architecture) about that system prior to its physical construction and implementation. This system knowledge (architecture) is described and represented by a set of views, which collectively reflect the concerns and the requirements of the stakeholders of that system. Since the data architecture (or data view) is common and important for information system development, the presence of an enterprise data viewpoint or resource as an element in the ESE component will provide common and necessary services. These services will facilitate the construction of the system data view with less effort, time and cost. This will make the constructed system the data view conformant and compatible with enterprise data viewpoint/resource.

As another example, using the C4ISR AF for developing future systems, architecture teams need to use certain supporting elements or reference resources to generate (architecture) products grouped into three main views, that is, Operational view, System view and Technical view.

In the architecture development for future systems, the main role the ESE component plays, is to support SACP in the generation of architecture relevant to future systems and to the evolution of existing systems.

I. ESE Development

Selecting and developing a proper set of ESE for a particular organisation is important since different organisations may have different requirements in architecture practice. Some elements that are considered as part of ESE are well defined as a set of architectures, because of their relevance in the context of systems development. For instance the domain architectures of the Meta Group approach and the Common Building Blocks (LISI, TAFIM, UJTL and so on) of the C4ISR AF.

As suggested by some architectural approaches (Meta Group, MSF and Zachman), there is generic guidance for developing certain elements or viewpoints that are commonly used for information systems development includeing:

- *Vision Statement* — What are the main functions of a particular resource/viewpoint — Purpose of the resource existence.

- *Principles* — Foundations on which the objectives of the resource can be achieved.

- *Guidelines* — Procedures required to establish and implement the principles.

- *Best Practice* — How best to implement the steps in the procedures.

- *Standards* — Best practice rely on national and international standards for their implementations.

- *Products* — to support implementation of best practice in compliance with standards.

The terms above form the foundation for efficient and effective development of ESE.

For large organisations committed to use of architecture to support their future development, the decision on which supporting elements should be developed is critical. This is related to the decision on which frameworks or approaches will be applied. In addition, organisations need to decide whether the supporting elements defined by the frameworks or approaches chosen are sufficient and whether other specific elements need to be developed.

II. Use of ESE

The value of ESE can be proved only when they are used in practice. In order to efficiently and effectively use ESE, these elements should be developed as part of the Architecture Practice Supporting Environment (APSE) as architecture reference resources. The decision made by the organisation on which methodologies should be applied for developing new systems will also need to specify how ESE should be used as references to guide their architecture development.

6.3.2 System Architecture Construction Process (SACP)

Because of the different interests of projects, SACP is a general term for all activities that generate descriptive architecture products. This is regardless of whether they generate a complete set of all views or just a single view, and no matter whether the architecture will be used for system development or for other research/development interests such as planning, modelling and simulation. One of the main purposes of architecture practice is to let all these architecture development activities benefit from developed architecture resources including ESE and the Enterprise Architecture Repository, and to be supported by APSE.

System architecture construction can be carried out in many different ways depending on what interests or purposes the architecture is developed for and how it is developed — whether using any framework or approach.

Conducting SACP is the most common and traditional activity of architecture practice and has been supported by many architecture tools and guided by various methodologies. These forms of support and guidance will continue to play important roles in practice.

The architecture practice study does not repeat the efforts made by methodology developers in guiding architecture development. The purpose of discussing SACP in the architecture practice conceptual model is to show, in addition to use of certain architecture approaches, how this activity can be carried out in a more efficient and cost-effective manner through using the more-comprehensive services provided by APSE developed specifically for the organisation.

The main role that the SACP component can play, in the acquisition of systems and their knowledge, is in the generation of the architecture (knowledge) of future systems. The SACP can use the services and products, made available by the Enterprise Supporting Elements (ESE), to construct new system architecture. On the other hand, if the system architectural views do exist as reusable products, then SACP will use the EAR component to retrieve the reusable views.

6.3.3 Systems Architecture Acquisition Process (SAAP)

After three decades of IT applications development, various information systems exist and are continuously operating to support business within large organisations. Assume that ideally all these existing systems have been successfully developed under guidance of their design documents or architectures. They are likely to be "stovepipes" that have yet to be integrated to better support the organisation's business. Preserving the architectures of existing systems, or the associated architectures of existing systems, realises the current picture role of architecture. These preserved "current pictures" of existing systems are knowledge assets of the organisation, which are not only useful but also extremely important for developing new or changing existing systems.

There are four issues that arise in preserving these knowledge assets for the entire organisation or enterprise:

1) Whether architectures have been developed and maintained up-to-date for all existing systems.
2) Whether there is a need to re-describe some systems that are not represented in any architecture.
3) Whether further efforts are required in order to get a complete picture of the "big system" (whole enterprise), or a system of systems if all architectures of existing systems were developed at a project level for individual systems.
4) How issues 2 and 3 can be addressed.

Traditional architecture-related activities and methodologies have not addressed these issues. There has been a sub-task of the Architecture Practice Study carried out jointly by IAG of JSB, DSTO and Monash University, aimed at addressing the above issues by introducing the concept of "Systems Architecture Acquisition at the Enterprise Level", which is achieved by carrying out the following activities:

- Maintaining synchronised evolution of both the existing system and its associated architecture.
- Translating the existing system's associated architecture from system-specific descriptions to enterprise-specific description using a unified architecture description. This process can be called the systems architecture acquisition at the enterprise level. Its result is the enterprise systems architecture. This enterprise description of systems should be of a standard format that can be understood, accessed and reused across the enterprise.
- Saving the description in the Enterprise Architecture Repository (EAR).

More detailed discussion of these issues is included in the technical reports of the joint DSTO-Monash project.

The component SAAP in the architecture practice conceptual model is implemented through carrying out the three main activities mentioned above.

6.3.4 Enterprise Architecture Repository (EAR)

Another important component of APSE is EAR. We envisaged that this repository will consist of three types of architectures stored in three sub-repositories:

1. *Blueprint repository* — this sub-repository will contain architectural description of all systems as blueprints (planned architecture). These architectural descriptions are essential prior to the development and implementation of new systems. It is possible to generate more than one architecture for the same planned system, which are called "to be" architectures.

2. *Associated repository* — this sub-repository will contain architectural descriptions of each existing system, which is the current picture, or "as-is" architecture. These architectural descriptions are essential prior to the understanding of individual existing systems.

3. *Systems architecture repository* — From the enterprise point of view, this sub-repository is more important than the first two. The reason for its importance is

that, first, it will contain architecture (knowledge) for each of the existing systems, described in a unified notation, which will make it more accessible, understandable and reusable across the enterprise; second, it will support generation of "as-is" architecture of a system of systems (SOS). In other words, this repository can facilitate systems acquisition at the enterprise level.

One can easily notice that the first two sub-repositories can be achieved without major efforts of SAAP. The main outcome of SAAP, thus, is the systems architecture repository.

Another observation is that existing architecture practices have extensive experience in developing the first two repositories if they are just collections of individual architectures. However, their extensive experience does not include development of the third repository.

The third sub-repository is an active area of research. The main issues of designing and developing this sub-repository are also discussed in the technical reports of the joint DSTO-Monash project.

The main roles the EAR component will play in improving the acquisition of systems and their knowledge are:

- To support the System Architecture Construction Process (SACP) in generating and evolving architectures for future and existing systems respectively. This support is achieved by allowing the SACP to easily access the reusable products (systems knowledge) preserved in EAR.
- To preserve and maintain the generated architectures from both SACP and SAAP.

6.4 Principles of the Recommended Architecture Practice

As mentioned above, architecture practice started in large organisations in an unplanned manner. It continues and becomes more influential as more architecture products are generated by various activities. There are some questions that may be confusing: What is the relationship between architecture practice and individual architecture development activities? What should the main outcome of architecture practice be?

The answer to the first question is twofold:

1) Any architecture activity is part of architecture practice.
2) A disciplined and well-managed architecture practice can let architecture activities be carried out in a more productive and cost-effective manner and produce better outputs — including architecture products.

To answer the second question, let us first recall a remark from Section 2:

"It is time, thus, for large organisations to examine at what level their IT development capability or future development capability is currently, whether it could actually meet the

requirements of the organisation through delivering quality IT capability to support business, and to find out how this IT development capability or future organisation development capability can be improved if it is not satisfied today."

Rather than generating any individual architecture product that is produced by a particular activity, the main outcome of a *disciplined* and *well-managed* architecture practice is to improve the capability of the organisation in IT practice or future organisation development by providing the organisation with an integrated architecture capability.

Different organisations have different requirements of the IT development capability. Part 2 of the Phase I Technical Report from the Architecture Practice Study task discusses specifically the requirements of IT development capability for the Australian Defence Organisation (ADO) and how the architecture practice can support improvement of this capability.

Based on the context and complexity analysis of architecture activities, we can use the architecture practice conceptual model to derive main principles of a disciplined architecture practice for large organisations.

Principle 1 — Plan and develop architecture and methods in context.
Principle 2 — Coordinate use of architecture frameworks/approaches.
Principle 3 — Realise fully the value of architecture for both descriptive and supporting products.
Principle 4 — Preserve architecture value over time.
Principle 5 — Integrate architecture capabilities and services.

These principles mark the distinction between a *disciplined* and *managed* architecture practice and an *unplanned* and *uncoordinated* architecture practice. The discussions and analysis given in the Sections 4, 5 and 6 aimed to indicate some approaches to implementation of these principles.

If these principles can be followed by an organisation in its architecture practice, *an integrated architecture capability* should be generated to effectively support the improvement of future organisation development capability of which the IT development capability is part. Such an integrated architecture capability should include:

- Architecture resources that include various architecture products, both descriptive and supporting, developed on the basis of a well-established architecture lexicon.
- Architecture-based functions/services, including generic and specific, which can realise effective use of architecture for various purposes in all aspects of future organisation development.
- Well-defined and well-coordinated architecture development processes guided by well-chosen methodologies.
- An Architecture Practice Supporting Environment that provides not only solutions for architecture resources management, but also a unified and integrated basis for architecture-based functions/services development.

6.5   Feature and Limitation Analysis of Architecture Frameworks

After introducing the architecture practice conceptual model and discussing the principles of a disciplined and well-managed architecture practice, we can further examine selected architecture frameworks or approaches in terms what issues are addressed by them and what issues remain.

Generally speaking, an architecture methodology or framework provides guidance in architecture development for a particular *sub-area* of the whole practice by defining a set of viewpoints and/or supporting elements and certain processes for producing certain types of architecture products. This is why there are many different architecture frameworks or approaches of which some are developed to address similar problems and some address quite different issues from others.

How big such a *sub-area* is depends on each methodology — supporting the range from only programming, or a single system development, to enterprise-wide development. Whether a methodology that claims to support enterprise-wide development is sufficient to address all needs of the organisation in architecture practice is an interesting question. The answer from the developer of the methodology might well be "yes". From our architecture practice study point of view, however, the answer is that depending on the nature of the organisation it may not be enough and there is also a need to examine its applicability. Since the size and nature of organisations vary, their requirements of architecture practice are quite different. A methodology that can successfully support or guide architecture practice for a small organisation in its specific development settings may have difficulty or sometime even be improper when it is applied to a large organisation in a quite different development setting.

As mentioned earlier, the enterprise supporting elements (ESE) are defined differently in many architecture frameworks or approaches due to their different focuses or objectives. The need to use multiple architecture frameworks can be observed when an organisation finds there is no such framework that can provide a complete set of ESE required by its practice and guidance to develop architecture products of interest to the organisation.

The principles of architecture practice discussed in Section 6.5 are *partially* shared by those architecture frameworks or approaches since they can, to a certain extent, support the implementation of some principles, such as planning and selecting some elements of ESE for their specific purposes and realising the value of architecture products generated through using the frameworks.

Combining the analysis above with the methodologies review of Section 4, we now establish a common basis to examine and compare different architecture frameworks or approaches. The examination starts with the following questions:

- What main architecture issues does a framework or approach address?
- How does it deal with the concept of ESE?
- How does it deal with the architectures of existing systems or systems architecture acquisition at the enterprise level?

- How are the architecture issues addressed by the framework related to architecture issues and products, which are *not* covered by the same framework?
- What are architecture capabilities (a single product, a set of both descriptive and supporting products, management solutions, tools or practice supporting environments) that can be delivered by the framework?

However, the examination of a framework is a form of subjective evaluation depending on personal interests and understanding. Instead of evaluating in details all those frameworks or approaches, we suggest that organisation with interests in those methodologies perform their own evaluation through combining these questions with their specific interests.

7. Architecture Practice Improvement

Each of the four main components of the practice (SACP, ESE, SAAP and EAR) has certain roles to play, in dealing with various architectures or knowledge regarding IT practice and future organisation development. How well these architecture practice capabilities are implemented directly reflects the level of an organisation's ability to handle architecture-type organisational knowledge. Through assessing the capability level of knowledge (architecture) acquisition and use, organisations can assess their current architecture practice and whether or not their practice requires improvements. This section discusses how architecture practice can be assessed and through improvement of individual components and combination of these components can help to improve the capability level of knowledge acquisition and use.

Assessing the level of architecture practice for a particular organisation requires examination of three main facts:

- How these components of architecture practice are covered by architecture activities in the past and currently in terms of whether the activities or their outputs are considered as part of any of these components.

- How well the activities have been carried out to deliver the capabilities of the components or how well individual components have been implemented.

- How the capabilities arising from these components have been integrated or how the value of individual architecture products has been realised throughout the architecture practice.

Through such an examination, the organisation can see clearly the current architecture efforts and identify the need for improvement.

7.1 Component Improvement Level

The Component Improvement Model in Figure 7-1 depicts five improvement levels that each of the four main components can be assessed against. There is a common requirement for all these levels that skilled and well-trained professionals are required to perform practice. The criteria that will be used to measure the improvement level for each of the four main components are detailed below:

- **Level 1 — Defined and Documented**

If a particular component, within the proposed practice, is assessed as "defined and documented", then the assessed component is considered to have reached Level 1.

- **Level 2 — Available and Accessible**

If a particular component, within the proposed practice, is assessed as "available and accessible", then the assessed component is considered to have reached improvement level 2.

- **Level 3 — Compliant with Standards and Best Practice**

If a particular component, within the proposed practice, is assessed as "compliant with standards and best practice", then the assessed component is considered to have reached improvement Level 3.

- **Level 4 — Integrate-able with Other Components**

If a particular component, within the proposed practice, is assessed as "integrate-able with other components", then the assessed component is considered to have reached improvement Level 4.

- **Level 5 — Subjected to Process Innovation**

If a particular component, within the proposed practice, is assessed as "subjected to process innovation", then the assessed component is considered to have reached improvement Level 5.

Component Improvement

Level 5

Level 4

Level 3

Level 2

Level 1

Subjected to
Process
Innovation

Integrateable with other
components

Compliant with
Standards and Best Practice

Available and Accessible

Defined and Documented
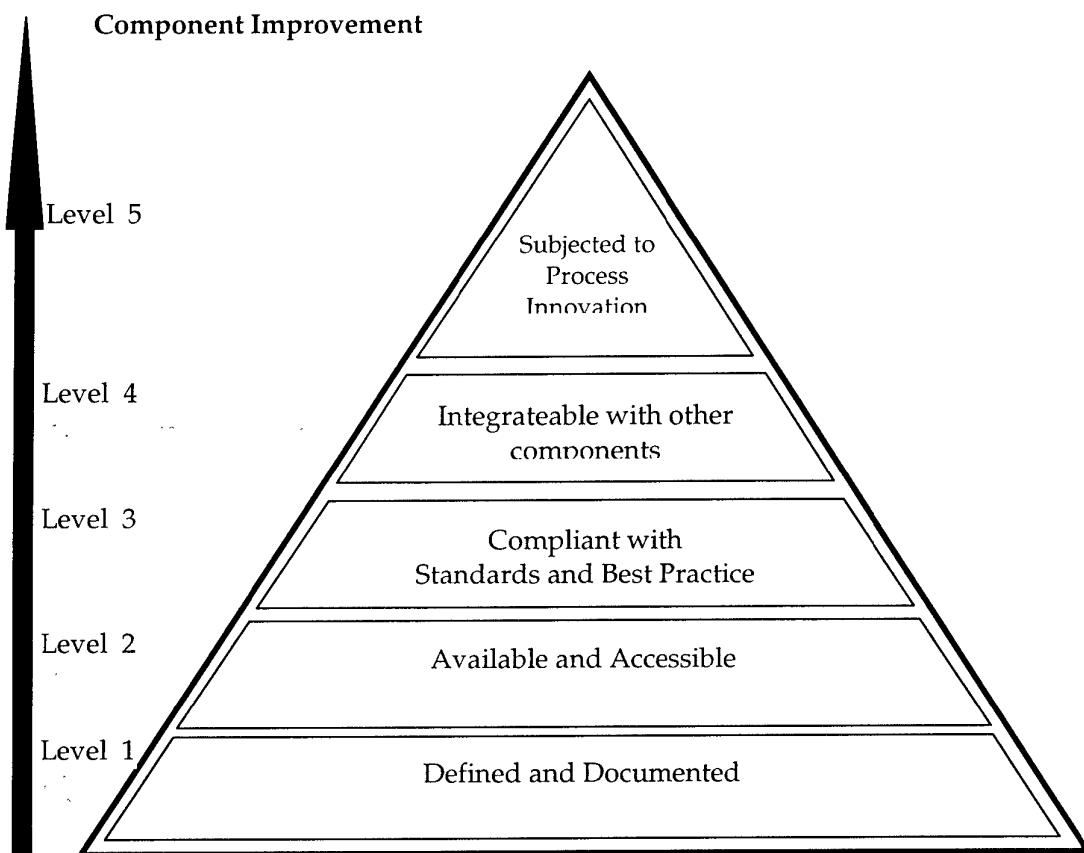
Figure 7-1 Component Improvement Model

Process innovation, after reaching Level 4, is introduced to further improve or enhance processes of architecture-related activities including production, management and use through applying better methodologies and developing advanced supporting tools and environments.

7.2 Knowledge Acquisition Improvement Level

Different architecture practices lead to different levels of organisational knowledge acquisition and use in supporting future development. The Knowledge Acquisition Improvement Model shown in Figure 7-2 depicts four improvement levels which can be reached when different components of architecture practice are implemented at Level 2 or above of their individual improvement. These four levels of improvement are also based on how well the four main components are integrated and managed within the practice. Using the proposed architecture practice as a common basis, we outline in detail the criteria that can be used to measure each of four improvement levels, and the characteristic of the practice at each level.

Enterprise Knowledge Acquisition Improvement
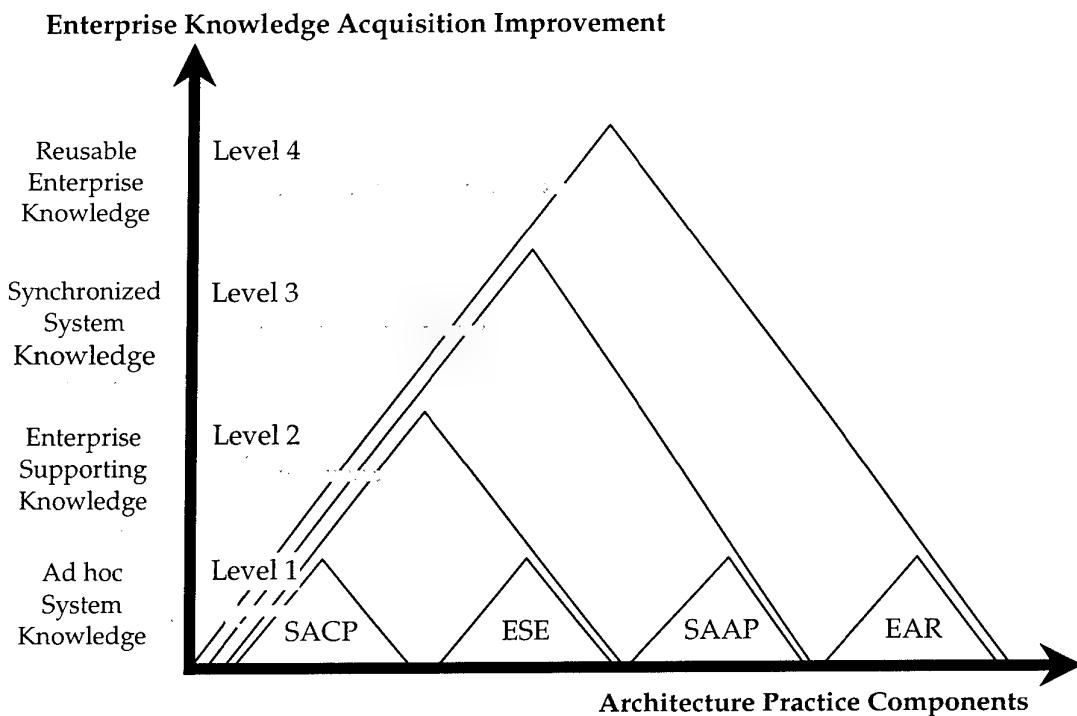

Figure 7-2 Knowledge Acquisition Improvement Model

- **Level 1 — Ad–hoc System Knowledge**

The knowledge acquired by the enterprise, through its architecture practice, is assessed as having reached Level 1, if the architectures (Knowledge) of its systems are acquired in an ad-hoc manner. The practice at this level is characterised by the following:

➢ No integration exists between any of the four components. System knowledge is totally created by SACP in total isolation from the other components.

➢ The created system knowledge (architecture) is considered to be ad–hoc, inconsistent in its representation, and costly and time–consuming in development due to lack of methodology supports for integration between the SACP and ESE.

➢ The created system knowledge (architecture) is mainly described for project development and not preserved for reuse at the enterprise level.

- **Level 2 — Enterprise Supporting Knowledge**

The knowledge acquired by the enterprise, through its architecture practice, is assessed as having reached Level 2, if the architectures (Knowledge) of its systems are acquired using ESE services. The practice at this level is characterised by the following:

> Integration exists between two of the practice's components: SACP and ESE. The effectiveness of this integration depends on the improvement level each of these two components has reached (see Figure 7-2). The higher the improvement level, the better the effectiveness of the integration.

> System knowledge is created by SACP, with total reliance on the common services provided by the ESE component.

> The created system knowledge is considered to be planned, consistent in its representation, and less costly and time–consuming in development due to the integration between the SACP and ESE through introducing well-defined process guidance or applying proper methodologies/frameworks.

> The created system knowledge (architecture), however, is neither described nor preserved specifically for reuse at the enterprise level.

- **Level 3 — Synchronised System Knowledge**

The knowledge acquired by the enterprise, through its architecture practice, is assessed as having reached Level 3 if the architectures of its systems are acquired in a synchronised manner. The practice at this level is characterised by the following:

> Integration exists between three of the practice's components: SACP, ESE and SAAP.

> The created system knowledge (architecture or blueprints) is considered to be planned, consistent in its representation, and less costly and time–consuming in development due to the integration between the SACP and ESE through introducing well-defined process guidance or applying appropriate methodologies/frameworks.

> The systems knowledge (architectures of existing systems) is considered to be synchronised, consistent and is kept as a current picture due to the action of SAAP.

> The preserved and synchronised systems knowledge is available for reuse.

- **Level 4 — Reusable Enterprise Knowledge**

The knowledge acquired by the enterprise, through its architecture practice, is assessed as having Level 4 if the architectures (knowledge) of its systems are acquired in a reusable manner. The practice at this level is characterised by the following:

> Integration exists between all of the practice's components: SACP, ESE, SAAP, and EAR.

> The "Synchronised System Knowledge" created in Level 3 is now improved and maintained and becomes "Reusable System Knowledge" due to the services provided by EAR.

> ➤ The created system knowledge (architecture or blueprints) for new systems is considered to be planned, consistent in its representation, and developed in an efficient and cost-effective manner due to the integration among the SACP, ESE and EAR supported by well-defined process guidance and appropriate methodologies and/or frameworks.

> ➤ There are continuous efforts made in SAAP to achieve the complete acquisition of systems architecture at the enterprise level and make EAR an evolve-able organisational asset.

Note the characterizations given above are those important and fundamental activities carried out and products generated from the practice. It is not only possible but also important in developing more advanced architecture capabilities (modeling, visualization, simulation and architecture analysis and inference) that this be based on the resources and services from ESE and EAR, integrated them APSE.

Since the level of architecture knowledge development, acquisition and use is directly determined by the improvement level of architecture practice, Figure 7-2 can also be seen as an *architecture practice maturity model*.

7.3 Architecture Practice Improvement Guidance

To facilitate the improvement of architecture practice, this subsection provides general guidance for moving from one level to another in the architecture practice maturity model.

- **High Level Guidance for Advancing the Practice from Maturity Level One to Maturity Level Two**

To advance the maturity of the enterprise architecture practice from Level 1 to level 2:

> ➤ Use viewpoint development approach (see Section 6.4) to enhance existing viewpoints or ESE and to develop additional elements that are necessary for meeting stakeholder concerns and supporting architecture (views) construction.

> ➤ Develop services (improved accessibility and usability) from each viewpoint that are relevant for use as input to the System Architecture Construction Process (SACP).

> ➤ Link relevant services with adequate methodologies (such as C4ISR AF) used in SACP to aid in the construction of views (blueprint).

> ➤ Preserve the blueprint in a repository.

> ➤ Enhance the training and education received by professional staff to provide quality support for SACP and ESE.

- **High Level Guidance for Advancing the Architecture Practice from Maturity Level 2 to Maturity Level 3**

To advance the maturity of the enterprise architecture practice from Level 2 to Level 3:

➤ Identify variations in consistency/synchronisation of architectures (knowledge) of its systems between pre-implementation (blueprint) and post-implementation (associated).

➤ Develop processes and technology to eliminate variances and update the associated architectures (or current pictures) of the existing systems if necessary.

➤ Preserve the current pictures in a repository.

➤ Enhance the training and education received by professional staff to provide quality support for ensuring that consistency of architectures is maintained.

- **High Level Guidance for Advancing the Practice from Maturity Level 3 to Maturity Level 4**

The steps below offer a high level guidance to advance the maturity of the enterprise architecture practice from level three to level four:

➤ Identify which views of each existing system will need to be translated to enterprise level.

➤ Use the System Architecture Representation Process (SAAP) to perform the translation.

➤ Preserve the translated views (Systems Architecture that is the current picture of SOS) in a repository.

➤ Enhance the training and education received by professional staff to provide quality support for ensuring the enterprise knowledge of existing systems is consistent, accessible, understandable and shareable.

8. Architecture Practice Critical Success and Failure Factors

In developing and implementing architecture practice, the critical success factors that an organisation should adhere to, include, but are not limited to:

- Commitment by senior management ensures consensus on the definition and roles of architecture, and communicates the definition and importance of architecture across all levels of the organisation.

- Identification and classification of Information Systems' stakeholders and their concerns.

- Management of the architecture practice is the responsibility of the organisation and a high-level practice model is defined for overall guidance.

- Proper use is made of capabilities in architecture from external agencies.

- A group is established to develop an Architecture Practice Supporting Environment in the organisation. This environment caters for the concerns and requirements of existing and future stakeholders through delivering adequate architecture-based capabilities to different areas.

- Group/s to develop and support System Architecture Construction Process and Systems Architecture Acquisition Process are established in the organisation. The system architecture construction process imports knowledge from the architecture–based development environment while the systems architecture acquisition process exports knowledge to that environment.

In developing and implementing an architecture, the critical failure factors that an organisation should avoid, include, but are not limited to:

- There is a lack of commitment by senior management to ensuring and communicating consensus on the definition and importance of architecture.

- Information System stakeholders and their concerns are not adequately identified and classified.

- Architecture development efforts are made in an unclear context.

- The responsibility for developing and implementing architecture practice is left completely to vendors.

- Expectations are too high in terms of what a particular architecture can deliver

- There is a lack of coordination in developing and supporting APSE, SACP and SAAP in the organisation.

9. Conclusion

This report has proposed and discussed architecture practice thinking in large organisations and how this thinking can contribute to future development of these organisations. The focus is on the following aspects:

- Context, settings and complexity of architecture practice.
- Relevance and connections among architecture products, activities and methodologies.
- Problems analysis.
- Review and evaluation of architecture methodologies.
- Principles study of architecture practice.
- Recommended conceptual model of disciplined practice, its principles and its capability.
- Architecture practice improvement and maturity.

The current report presents only initial findings from the study. There are many important issues in architecture practice that remain to be addressed. Such issues include how to define and develop a communications system or a formal notational system [Boar, 1999] that allows an architecture practice community to visualise and understand architectures in a consistent and standard manner.

In conclusion, this report lays the foundations for large organisations including the ADO to establish an architecture practice, as an inseparable part of IT management, based on developing and integrating the capabilities of the four components of the architecture practice conceptual model. The challenge is and will be to communicate this model and its potential benefits throughout ADO and gain acceptance by the majority of the concerned stakeholders from every corner in the organisation. It is through such a practice that the ADO can develop or evolve its information-based systems with reliability, adaptability and interoperability, thereby enabling the ADO to better develop new and evolve existing capabilities.

10. References

[AFRL, 1998) Air Force Research Lab. *Joint C4ISR Architecture Planning/Analysis System (Concept of Operations)*, 1998.

[Bass et al., 1998) Bass L., Clements P., and Kazman R., *Software Architecture in Practice*, Addison-Wesley Longman, USA, 1998.

[Boar, 1999] Boar B. H., *Constructing Blueprints for Enterprise IT Architectures*, Wiley Computer Publishing, 1999.

[CAWG, 1997a] C4ISR Architecture Working Group, *C4ISR Architecture Framework (Version 2.0)*, USA, 1997.

[CAWG, 1997b] C4ISR Architecture Working Group, *C4ISR Core Architecture Data Model, Version 1 (CADM)*, USA, 1997.

[CAWG, 1998] C4ISR Architecture Working Group, *C4ISR Levels of Information Systems Interoperability (LISI)*, USA, 1998.

[CSC, 1996] Computer Science Corporation, *Aligning Technology and Corporate Goals is Top Concern again in Annual CSC Survey — Networks Seen as Key Competitive Tool; Outsourcing Gaining Support.* 1996.
http://www.csc.com/about/news_stories/an_survey.html

[Chen, et al., 1998] Chen P., El-Sakka A., and Clothier J. *Context Analysis of Architecture Practice within Large Organizations*, Proceedings of the Australasian Workshop on Software Architectures, pp 2—13, Melbourne, Australia, (1998),

[DISA, 1996a] Defence Information Systems Agency *US DoD, Defence Information Infrastructure (DII), Common Operating Environment (COE).* USA, 1996.

[DISA, 1996b] Defence Information Systems Agency, *US DoD, Technical Architecture Framework for Information Management (TAFIM)*, USA, 1996.

[El-Sakka, et al., 1999] El-Sakka A., Chen P., and Clothier J., *Knowledge Acquisition Improvement through Enterprise Architecture Practice*, Proceedings of the Software Engineering Australia 1999 (SEA'99), Canberra, Australia, 1999.

[GG, 1997] Gartner Group Dataquest (1997), *Managing High Turnover*, 1997.
http://www.dqindia.com/jan1597/3ia1211101.html

[Horowitz, 1996] Horowitz B. M., *The Importance of Architecture in DoD Software, Guidelines for Successful Acquisition and Management of Software-Intensive Systems*, USA, 1996.

[IAWG, 1998] IEEE Architecture Working Group), *IEEE Recommended Practice for Architectural Description, IEEE Std 1471, Draft Version 3.0*, USA, 1998.

[ISO, 1996] Information Standards Organisation (1996), *Open Distributed Processing – Reference Model (ODP-RM), ISO/IEC DIS 10746-2, and ISO/IEC DIS 10746-3*, USA, 1996.

[ISO/IEC 9126] Delta Software Engineering, Quality Characteristics, ISO/IEC 9126 Information Technology – Software Product Evaluation – Quality Characteristics and Guidance for their Use, www.delta.dk/se/scope/intro/b1.htm.

[ITO, 1997] Information Technology Office DARPA, *Evolutionary Design of Complex Software, Technology Showcase*, UK, 1997.

[Lockhead, 1996] Lockheed Martin, Tactical Defence Systems, *Organization Domain Modeling (ODM) Guidebook (Version 2.0)*, USA, 1996.

[Maier, 1997] Maier, M. W., *Architecting Principles for Systems-of-Systems*, USA, 1997. http://www.infoed.com/Open/PAPERS/systems.htm

[Martin *et al.*, 1994] Martin E. W., DeHayes D. W., Hoffer J. A. and Perkins W. C. (1994), *Managing Information Technology*, Prentice Hill, International Edition, USA.

[Meta Group, 1999] Meta Group, *Enterprise Architecture Strategies (EAS)*, Meta Delta, USA, 1999.

[Microsoft, 1999a] Microsoft, *Glossary and Acronym*, USA, 1999. http://www.microsoft.com/hwdev/glossary.htm

[Microsoft, 1999b] Microsoft, *Microsoft Solutions Framework (MSF)*, USA, 1999. http://www.microsoft.com/solutionsframework

[NRC, 1999] National Research Council, *Realising the Potential of C4I: Fundamental Challenges*, National Academy Press (Prepublication Draft), USA, 1999.

[New Denver Airport, 1994] New Denver Airport, *Impact of the Delayed Baggage System (Briefing Report, 10/14/94, GAO/RCED-95-35BR)*, USA, 1994. www.bts.gov/NTL/DOCS/rc9535br.html

[OIC, 1997] Object Ideas Corporation *Enterprise Architecture Modelling*, USA, 1997. http://www.object-ideas.com/eamdl/tsld001.htm

[Pirkola, 1995] Pirkola G., *An Information Technology (IT) Architecture for Information Technology Division*, University of Michigan, USA, 1995. http://www-personal.umich.edu/~gpirkola/IT_arch_doc/Phase1Doc.html

[SEI, 1999] Software Engineering Institute, *Definitions on Architecture*, USA, 1999. http://www.sei.cmu.edu/architecture/definitions.html

[US DoD, 1996] US DoD, *Guidelines for Successful Acquisition and Management of Software Intensive Systems, Version 2.0. ISBN 0-9664169-1-0*. USA, 1996.

[Zachman, 1996] Zachman J., *Enterprise Architecture: The Issue of the Century*, USA, 1996. http://www.zifa.com/zifajz01.htm.,

Context Analysis and Principles Study

Phase 1 - Client Report (Part 1)
Architecture Practice Study

P. Chen and A. El-Sakka
(DSTO-CR-0151)

DISTRIBUTION LIST

Number of Copies

AUSTRALIA

DEFENCE ORGANISATION

Task Sponsor:

DGC3ID	1
DDIEAO	1

S&T Program

Chief Defence Scientist)	
FAS Science Policy)	1 shared copy
AS Science Corporate Management)	
Director General Science Policy Development	1
Counsellor, Defence Science, London	Doc Control Sheet
Counsellor, Defence Science, Washington	Doc Control Sheet
Scientific Adviser to MRDC Thailand	Doc Control Sheet
Scientific Adviser - Policy and Command	1
Navy Scientific Adviser	Doc Control Sheet and distribution list
Scientific Adviser - Army	Doc Control Sheet and distribution list
Air Force Scientific Adviser	1
Director Trials	1

Aeronautical & Maritime Research Laboratory

Director	1

Electronics and Surveillance Research Laboratory

Director	Doc Control Sheet and distribution list
Chief Information Technology Division	1
Research Leader Joint Systems	1
Research Leader Command & Control and Intelligence Systems	1
Research Leader Military Information Enterprise	1
Head, Military Systems Synthesis Group	1
Head, Systems of Systems Group	1

Head C3I Systems Concepts Group	1
Head, C2 Australian Theatre	1
Head, Distributed Systems Group	1
Head, Organisational Change Group	1
Head, Trusted Computer Systems Group	Doc Control Sheet
Head, Systems Simultion and Assessment Group	Doc Control Sheet
Head, C3I Operational Analysis Group	Doc Control Sheet
Head, Information Management and Fusion Group	Doc Control Sheet
Head, Human Systems Integration Group	Doc Control Sheet
Head, Advanced Network Integrity Group	Doc Control Sheet
Publications and Publicity Officer, ITD/Executive Officer, ITD	1
Pin Chen, IA Group, JSB	1
Abdel El-Sakka, IA Group, JSB	1
Gary Bulluss, IA Group, JSB	1

DSTO Research Library and Archives

Library Fishermens Bend	Doc Control Sheet
Library Maribyrnong	Doc Control Sheet
Library Salisbury	1
Australian Archives	1
Library, MOD, Pyrmont	Doc Control Sheet
US Defence Technical Information Center	2
UK Defence Research Information Centre	2
Canada Defence Scientific Information Service	1
NZ Defence Information Centre	1
National Library of Australia	1

Capability Systems Staff

Director General Maritime Development	Doc Control Sheet
Director General Land Development	1
Director General C3I Development	1
Director General Aerospace Development	Doc Control Sheet

Navy

SO (Science), Director of Naval Warfare, Maritime Headquarters Annex, Garden Island, NSW 2000	Doc Control Sheet

Army

ASNSO ABCA, Puckapunyal	4
SO (Science), DJFHQ(L), MILPO Enoggera, Queensland 4051	Doc Control Sheet

Intelligence

DGSTA, Defence Intelligence Organisation	1
Manager, Information Centre, Defence Intelligence Organisation	1
DADFISS	1

Defence Information Systems Group
DGCIPP-DIS 1
DIAS 1

Defence Acquisition Organisation
DGCOMMS 1
DGCSS 1
DSWAR 1

Joint Logistics Systems Agency
DLSD 1
DD AMDD 1
LEAP Project Manager 1

Corporate Support Program
Library Manager, DLS-Canberra Doc Control Sheet

Universities and Colleges
Australian Defence Force Academy
Library 1
Head of Aerospace and Mechanical Engineering 1
Deakin University, Serials Section (M list)), Deakin University Library,
Geelong, 3217 1
Senior Librarian, Hargrave Library, Monash University Doc Control Sheet
Librarian, Flinders University 1

Other Organisations
NASA (Canberra) 1
AusInfo 1
State Library of South Australia 1
Parliamentary Library, South Australia 1

<div align="center">

OUTSIDE AUSTRALIA

</div>

Abstracting and Information Organisations
Library, Chemical Abstracts Reference Service 1
Engineering Societies Library, US 1
Materials Information, Cambridge Scientific Abstracts, US 1
Documents Librarian, The Center for Research Libraries, US 1

Information Exchange Agreement Partners
Acquisitions Unit, Science Reference and Information Service, UK 1
Library - Exchange Desk, National Institute of Standards and
Technology, US 1
National Aerospace Laboratory, Japan 1
National Aerospace Laboratory, Netherlands 1

Authors 10
SPARES 5

Total number of copies: 82

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA		1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)	

2. TITLE Context Analysis and Principles Study Phase 1 – Client Report (Part 1) Architecture Practice Study	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)

4. AUTHOR(S) P. Chen and A. El-Sakka	5. CORPORATE AUTHOR Electronic and Surveillance Research Laboratory PO Box 1500 Salisbury SA 5108

6a. DSTO NUMBER DSTO-CR-0151	6b. AR NUMBER AR-011-438	6c. TYPE OF REPORT Client Report	7. DOCUMENT DATE October 2000

8. FILE NUMBER 011-439	9. TASK NUMBER JNT 99/017	10. TASK SPONSOR DGC3ID	11. NO. OF PAGES 79	12. NO. OF REFERENCES 30

13. DOWNGRADING/DELIMITING INSTRUCTIONS URL ON THE WORLD WIDE WEB: http://www.dsto.defence.gov.au/corporate/reports/DSTO-CR-0151.pdf	14. RELEASE AUTHORITY Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, SALISBURY, SA 5108

16. DELIBERATE ANNOUNCEMENT

No limitation

17. CASUAL ANNOUNCEMENT	Yes

18. DEFTEST DESCRIPTORS
Computer architecture
Information systems
Systems integration
Interoperbility

19. ABSTRACT

Large organisations are becoming increasingly dependent on the reliability, adaptability and interoperability of information-intensive systems to conduct their business operations successfully and profitably. These qualities are now prerequisite for organisations' survival in the ever-changing national and international markets. For the last two decades, architecture, despite its diverse definitions, has been considered by Information Technology (IT) practitioners and researchers, as playing critical roles during the life cycle of systems and their infrastructure. The roles include, firstly, providing a sound foundation on which quality information-intensive systems are developed or evolved, secondly, capturing the necessary knowledge to aid in the understanding about these systems, and thirdly, guiding the development of enterprise infrastructure needed to support the creation of such knowledge. This report concludes that large organisations, including the Australian Defence Organisation (ADO), must commit themselves to establishing and managing architecture as an inseparable practice embedded within their overall IT management and practice. Such recommendation, once implemented, will ensure that the identified roles of architecture are performed, sustained and continually improved. To pave the way towards this objective, a conceptual model is introduced to help in the planning and guidance of the development and incorporation of architecture practice within ADO.